LEVEL II

2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD A109550

DTIC
JAN 13 1982
S
E

THESIS

SIMULATION OF DISMOUNTED INFANTRY
COMBAT IN URBAN TERRAIN

by

Bradford G. Loo

September 1981

Thesis Advisor: S. H. Parry

Approved for public release, distribution unlimited

82 01 12 028

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A109	3. RECIPIENT'S CATALOG NUMBER 550
4. TITLE (and Subtitle) Simulation of Dismounted Infantry Combat in Urban Terrain		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bradford G. Loo		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1981
		13. NUMBER OF PAGES 170
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

combat model
urban combat model
infantry combat model

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis presents the foundation for a stochastic simulation model that will represent dismounted infantry combat in urban terrain. The Simulation of Tactical Alternative Responses, STAR, ground-air combat model, which represents the parent simulation program for this proposed representation of urban combat, will be discussed in this document. Two state of the art urban combat models, MOBACS

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED 251450
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

and URBWAR, will also be discussed. These two models were developed under contract for the United States Army but were never fully utilized as analytical tools. Certain aspects of these models lead to insightful approaches to the modelling of urban combat and these insights were instrumental in the development of this model. The major portions of this model are the representation of the terrain and the procedures for determining line of sight. Other areas of city fighting, e.g. movement and communications, were not modelled but will be discussed as future enhancements to this model.

Dist
A

Approved for public release, distribution unlimited

Simulation of Dismounted Infantry Combat in
Urban Terrain

by

Bradford C. Loo
Captain, United States Army
B.S., United States Military Academy, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1981

Author: ____________________

Approved by: James K. Hartman
Thesis Advisor

S H Parry
Co-Advisor

Kneale T. Marshall
Chairman, Department of Operations Research

W M Woods
Dean of Information and Policy Sciences

ABSTRACT

This thesis presents the foundation for a stochastic simulation model that will represent dismounted infantry combat in urban terrain. The Simulation of Tactical Alternative Responses, STAR, ground-air combat model, which represents the parent simulation program for this proposed representation of urban combat, will be discussed in this document. Two state of the art urban combat models, MOBACS and URBWAR, will also be discussed. These two models were developed under contract for the United States Army but were never fully utilized as analytical tools. Certain aspects of these models lead to insightful approaches to the modelling of urban combat and these insights were instrumental in the development of this model. The major portions of this model are the representation of the terrain and the procedures for determining line of sight. Other areas of city fighting, e.g. movement and communications, were not modelled but will be discussed as future enhancements to this model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

"Opposing forces offensive doctrine, with it's emphasis on speed, determines that built-up areas should be avoided. However, when this is not possible, the opposing forces will then aim to seize towns and built-up areas by surprise attacks from the line of march. . . Rapid thrusts are made to seize the most important objectives and streets, to split the area into isolated pockets of resistance, and then destroy them piecemeal."

- FM 30-102

"Taking into consideration the heavy economic concentration with the congested areas . . . the economic efficiency of the Federal Republic would be hard hit by the loss or the destruction of even a few of the smaller areas. Therefore, we will not be able to concede these areas to an invading force . . . This means that in case of an attack against us, there will be heavy fighting everywhere in and for our built-up areas."

- MAJ Hans A. Kratz ("Combat in Built-up Areas")

The scenarios of the modern battlefield and the battlefield of the next war are the basis for the continual development of an effective armed force to meet the challenges these scenarios project. The selection of the alternative which can best provide the most effective armed force is a formidable task for the military decision maker. It is often unclear what the battlefield of the future will entail, but irrespective of the dynamic shifting of military target priorities and the perceived threat objectives, the infantry soldier will always be involved in the next armed conflict.

The use of armed conflicts is the implementation of a nation's foreign policy. Whether the aggressor's goal is total domination of a nation or preservation of its own boundaries, the killing of people does not end the war. History has shown that wars are won by crippling or stagnating a nation's industries, the majority of which are in densely urban areas. The industrialization of a nation and the movement of people out of the inner city have promoted urban sprawl, linking once rural areas with the city through highways, industrial parks, residential areas and commercial buildings. No longer can the commander enjoy the luxury of bypassing the obstacles of modernization. Battles will have to be won in the urban areas if the war is to be won.

The military analyst presents to the decision maker a quantitative comparison of the feasible alternatives to a specific problem. The military decision maker has to choose among the alternatives a feasible solution to whatever question he is asked. An analytical tool available to the analyst is the computer simulation model. Of particular interest to the military analyst is the combined arms force-on-force simulation model such as the Simulation of Tactical Alternative Responses (STAR) model. With few exceptions, existing combined arms simulation models do not adequately represent urban warfare. This thesis presents the foundation for a combined arms simulation model that is to be used in the evaluation of dismounted infantry combat on urban terrain.

The state-of-the art combat models on urban terrain will be discussed in Chapter II.

Chapter III discusses the STAR ground-air combat model and will discuss in detail two particular aspects of this model that provided insight in the development of the urban model.

Chapter IV provides a general description of the urban model to include its capabilities, a description of the primary variables, the simulation process and an overview of the line of sight process in the model.

Some possible future enhancements to the model are discussed in Chapter V.

Appendix A defines the sets, entities, global variables, routines, and events in the model.

Appendix B explains the data required to set up the simulation.

Appendix C provides a detailed explanation of selected routines for this model.

Appendix D illustrates a sample run of the simulation with a detailed audit trail of a single combatant's actions throughout the simulation.

II. STATE OF THE ART

The purpose of this chapter is to present the available models that simulate combat on urban terrain. These models provide the type of resolution required for a detailed combat simulation.

The United States Army Infantry School has been tasked to conduct a Mission Area Analysis on Close Combat Light (Infantry). Because of the nature of their analysis, it was requested that a computer program that simulated combat in urban terrain be made available to them. Through their research and that of other United States Army agencies, no existing, running models were found in the inventory. Only two models were found both shelved in 1976, that simulated Military Operations in Urban Terrain (MOUT). These two models are:

1. MOBACS: A deterministic, high resolution model, which uses the BRL Bunker Vulnerability Model and the ARPA Infantry Firefight Model to simulate urban combat down to the squad/fire team level. This model was developed by Ketron Incorporated, 1400 Wilson Boulevard, Arlington, Virginia, and the urban terrain is "typical" of urban central Europe, i.e. the Fulda and Koblenz areas. The documentation for this model is contained in [Ref. 3] and [Ref. 4].

2. URBWAR: A probalistic, high resolution model that is designed to use both DYNTACS and ASARS modules to simulate urban combat up to the squad/fire team level. This model was developed by Rodman Laboratories, Rock Island Arsenal, Rock Island, Illinois, and creates the scenario in a five building complex. The documentation for this model is contained in [Ref. 5].

A. MOBACS

In November of 1976, Ketron Incorporated completed a five volume technical report (a portion of these volumes are [Ref. 3] and [Ref. 4]) under contract from the Defense Advanced Research Projects Agency (DARPA), which described an urban combat simulation. These documents supported a total package, to include the computer program, which provided the means to evaluate operational concepts, weapons and material systems, and the employment of forces in an urban area; specifically this model dealt in the Western European area. The model provided two different levels of simulation:

1. Force level operations - a Brigade-sized unit down to platoon level resolution and

2. Unit level operations - a company-sized unit down to squad/fire team level.

The force level operations simulation was an interactive game, with players inputting decisions and orders at the beginning of each combat period. Further detailed analysis of the players inputs and the unit sequences that followed

these decisions could be conducted in the unit level operations simulation, a closed game, with a further capability of sensitivity analysis of the combat results based on varying input parameters [Ref. 7; p. vi].

1. <u>Creating the Terrain</u>

The MOBACS model uses a two-step modelling process to develop the terrain. These steps are contained in two models, the terrain data base and the program terrain models, each of which is described below.

a. Terrain Data Base Model: This model describes the two methods the MOBACS model uses to represent the urban environment: (1) the quadratic surface representation, used to represent surface relief, area vegetation and natural features, and (2) the rectangular representation, used to represent buildings and linear vegetation.

The quadratic surface representation model, called the surface relief model, used a technique based on the model developed by G.W. Evans of the Stanford Research Institute, [Ref. 3; p. 34], which approximates the surface relief by a piecewise quadratic surface. The term surface relief, as used in MOBACS, is used to describe the earth surface elevations, area vegetation and natural features. The technique as is used MOBACS divides the total area of interest (10km by 10km) into 40 meter grid squares. The elevation of each corner of the 40 meter grid square is an input value and a quadratic surface equation within each grid rectangle is constructed, with the surfaces being continuous between grid squares [Ref. 3 pp. 34-36].

Buildings are represented as rectangles. The rectangles are described as line segments which are input as two data points; each point being an (x,y,z) coordinate.

b. Program Terrain Model: The Terrain Data Base Model, which is described as a data base model, was not conducive as a function-oriented terrain model, because the storage requirements (approximately 250,000 words) were too large, resulting in an increase in compile and run times to perform the functions of the model, e.g. lines of sight. As a result, the program terrain model was constructed as a reduced terrain model to be used in conjunction with the Terrain Data Base model. [Ref. 3; p. 37]

The Program Terrain Model constructed over the terrain area of interest, a density contour map of the area. In general, the concept was that the probable occurrance of a building can be predicted by a density surface of the value of the average square meters of structure area (local) per square meters of the sampling area (local). The ordered, interpolated sampling of this density value in the area of interest produced the desired surface area density function. A discussion of this procedure is contained in [Ref. 3], pages 37-38 and 55-61; the discussion is vague and does not explicitly state the rationale for this procedure. There was little consideration given to this type of terrain representation for this thesis because of the confusing nature of its construction and its utilization.

2. Terrain Related Functions

The terrain related functions of the model were:

a. Intervisibility - the determination of an expected fraction of all possible lines of sight between an observer and target,

b. Detection - the determination of a visual sighting of a target, given intervisibility,

c. Movement rate - determination of a unit rate of movement under a no-fire condition on the terrain,

d. Structure hardness - an input value used in the determination of structure penetration by a type of round and

e. Structured area target effects - the determination of the effect on and in structures given a hit (used with d above). [Ref. 3; p. 61]

The intervisibility process calculates a probability of line of sight, given

a. P(an element of structure or vegetation intersects the ground trace of the line of sight),

b. P(an element of structure or vegetation height intersects the line of sight)

c. P(terrain surface intersects the line of sight), and

d. The relative z-coordinate of the observer and target. This assumes, based on experimentation conducted within sample areas, that the intervisibility segment lengths

are distributed in accordance with a Poisson distribution with the parameter being a function of the density of the structure. [Ref. 3; p. 62]
The resulting equation for P(line of sight) is contained in [Ref. 3] page 65 with an explanation of all variables used to calculate this probability. Of particular interest in these calculation is the model's use of expected values of structure and vegetation height and probabilities of blockage due to these structures and vegetation. The structure of the terrain does not allow for a P(line of sight) equal to 0 or 1, but rather a P(line of sight), then a Monte Carlo draw to determine if line of sight exists.

The movement function is analogous to the intervisibility function, i.e. experimentation on the structure density surfaces showed that probable movement through an area is directly related to the average probability of passing a line of sight through the same area. The equations and assumptions are all outlined in [Ref. 3] pages 68-70 and will not be discussed here.

The development of the equations to determine penetration of a projectile through a structure can well be incorporated into any urban model. From [Ref. 3; pp. 71-74], the empirical formula for the depth of penetration of a non-fragmenting, concrete-piercing projectile into concrete is given by:

```
x = (( 222 * p * (d**.215) * v**1.5) / Y) + .5 * d
```

with:

x = penetration in inches

p = projectile weight in lbs/projectile cross-sectional area in inches squared.

d = projectile diameter in inches

v = velocity of projectile in ft/thousandth of a second

y = materials constant (a function of the compressive strength of the material)

Because smaller caliber ammunition has tendencies to flatten on impact, thus increasing their diameter and reducing their ability to penetrate, let: d' = d / k, with k being the constant associated with the projectile, i.e.

PROJECTILE TYPE	K
Armor piercing	1.0
Concrete piercing	1.0
Fragmenting steel case	1.1
Ball	1.7

Because the shape of the projectile has changed, let w = equivalent weight over the cross-sectional area based on the diameter d'. The equation for x thus becomes:

$$x = ((222*w*(d'**.215)*(v**1.5) / .25*\pi*y*(d'**2)) + .5*(d/d')$$

The quantity, k(m) = 222/.25*π*Y was calculated for different types of building materials. Letting k(m) = constant associated with material type m, these constants are as follows:

MATERIAL M	k (M)
Steel	.25
Stone	.94
Concrete	4.28
Masonry	5.51
Wood	17.30

The final equation for x thus becomes:

```
x = ((w*(d'**.215)*(v**1.5) / ((d'**2)*k(m))) +
                       .5*(d/d')
```

Given that a structure is x' inches thick, if $x > x'$, penetration has occurred.

When assessing personnel casualties on the inside of the building, given a projectile has penetrated, two assumptions have to be made if this formula is to be used:

a. The direction vector of the projectile on impact is equal to that after penetration, and

b. Empirically, based on the theory of a transfer of energy, the velocity of the projectile after penetration can be determined.

There are no empirical formulas which calculate the effects of penetration on a structure given a fragmenting projectile. Currently at AMSAA, there exists a very sparse data base, constructed by firing various types of U.S. projectiles at various types of structures. All projectiles were fired at a range of 250 meters at an angle of incidence of ninety degrees to the wall.

The types of projectiles were the (1) 60mm dual purpose, (2) 66mm LAW, (3) 90mm HEAT, (4) 105mm HEAT and HEP-T, and (5) 155mm HE. The structures were (1) 4/8 inch brick, (2) 12 inch brick, and 8, 10 and 16 inch reinforced concrete.

The data collected included:

a. Size of the hole made both outside and inside and

b. Direction and penetration capacity of the shrapnel and other projectiles created by the blast. This particular data was based on a set of panels being set up in a 16'x16'x8' room and any particle that penetrated any of these panels was considered to be incapacitating should a combatant be in the path of the particle.

Based on the collected data, AMSAA, using the random man model, was able to determine, that given a round penetrated a wall, the probability that a soldier was incapacitated could be calculated. This probability was probably based on the density of incapacitating projectiles passing over the coordination of the combatant.

To empirically fit all types of fragmenting projectiles against all types of structures is thought to be within the realm of reason by the personnel at AMSAA. To date, results of this empirical fitting have not been published.

B. URBWAR

This modelling project, completed in June 1975 by the Rodman Laboratory at Rock Island Arsenal, Illinois, is the model adopted by the Infantry School as the most likely

candidate for their study. The model is currently at TRASANA for validation and modification. This model is a stochastic, event-sequenced, small unit urban combat simulation written in the Fortran IV computer language. It models an urban area of five structurally different buildings and is capable of simulating the attack of defended buildings, the entering of defended buildings and the clearing of defended buildings. Each combatant cycles through three events during the simulation: detection, movement and firing.

1. Creating the Terrain and Combatants

The urban terrain is constructed by joining together a particular collection of parallelograms. This technique constructs the buildings and its rooms, floors and windows. The combatants are modelled as right circular cylinders which are formed in a subroutine when required, e.g. for lines of sight or casualty assessments. Otherwise the combatants are represented as (x,y,z) coordinates to the center of their heads.

2. Setting the Scenario

The initial positions of the combatants are input by the user. The combatants are broken into two separate forces, the attackers and the defenders. The attackers are further subdivided into a covering team and a rush team. The rush team is positioned outside a structure, but not in the field of view of the defense. The cover team and the

defense are situated within structures. The user has his choice of methods of inputting the defense. The first is to explicitly place them within the structure, or they can be placed randomly within the structure as if they were the "unsuspecting" force. [Ref. 3; pp. 3-2, A-3]

The scenario begins with each element receiving a movement time which is a random number, uniform (0,1). The user has the option of selecting who will have knowledge of the enemy and who will not. Based on the user's input as to who will have knowledge of the opposing force, this force will get assigned detection times which are uniform (0,1) random numbers. All times are in seconds of battle. Priority of events is first from lowest to highest event time, then by ranking the events: (1) detection, (2) movement and (3) firing in the event of a tie. [Ref. 5; p. A-3]

3. Terrain Related Functions

The detection event requires that each element be assigned "fire detection sectors" by input. [Ref. 5; p. 3-1] These sectors are designed to model fire discipline within either force. The sectors are ranked from most to least critical for the owning combatant and may be separate from, overlap into or be identical to other combatant's sectors. Elements can only search for and detect targets within these sectors. It is suggested that these sectors do not change dynamically as the combatant moves because the current version of the model can play only set scenarios which do not allow combatants to change buildings or objectives.

Detection can occur in one of two ways. The static non-firing target is detected based on the vision lobe theory. If the observer has line of sight, then a probability of detection is calculated based on the target-background contrast, presented area, range to the target and glimpse time. [Ref. 5; pp. 3-1, 7-3] If the observer is cued by a firing target, the Small Arms Requirement Studies (ASARS) model is used to determine probability of detection, accuracy of location and time to detection. [Ref. 5; p. 3-2]

The movement event produces three types of movement [Ref. 5; p. 3-2]:

a. Assault movement

b. Movements to obtain another search position

c. Movements due to suppression

Assault movement begins after the battle has started. All elements of the rush team move at a constant velocity in a straight line towards their first subobjective, i.e. an area not the defender's building. At this point the simulation stops. The model in its current form cannot dynamically move the combatants. Modifications of the movement logic are being made using the logic of the DYNTACS and ASARS models. The ASARS model allows the combatant to select routes based on paths of least resistance. This resistance is defined as a wall, building, or an area covered by the opposing force by either fire or observation. [Ref. 5; p. 3-6] A detailed narrative of the ASARS movement model is contained in [Ref. 6; vol IIA] Chapter VIA; the route selection process is on pages VI-A-22 of the same chapter.

Movement to a more advantageous detection position is done by simulating the observer crawling to the opposite side of the opening. If an observer has not detected a target by some set time, he is moved and continues his search.

Suppression movement occurs when the combatant is receiving fire or he needs to reload his weapon. Incoming fire forces the target to remain down for a fixed amount of time depending on the type of fire he is receiving. Reload times are weapon dependent.

The firing event determines if a target is hit, the number of times he is hit and whether or not he is a casualty.

The parameters for the event include the weapons performance capabilities, the range to the target, aiming and ballistics error, range estimation error, projectile impact velocity and mass and the tactical situation of the target, i.e. defense, rush or under cover. [Ref. 5; pp. 3-2, 3-3]

The probability of hit is calculated by either of two methods depending on the type of weapon employed.

a. If the firing weapon is semi-automatic the bivariate normal distribution model developed by AMSAA is used. If the firing weapon is automatic but as a light recoil or is on a stable platform, this model is also used.

b. For other automatic weapons, if the first round is distributed differently than all succeeding rounds, the tri-variate normal distribution is used. [Ref. 5; p. 3-3]

4. Shortcomings of the Model

The shortcomings of the URBWAR model were discussed in length by the authors of the model; the obvious one being the lack of dynamic movement of the combatants. Other areas discussed that are very germaine to any urban combat model are the lack of a validated data base or empirical equations for [Ref. 5; pp. 7-1 to 7-4]:

a. Penetration of materials by rounds,

b. Ricochet effects,

c. Aiming accuracy against fleeting targets and

d. Detections of targets inside structures.

C. EVALUATION OF MOBACS AND URBWAR

The two urban models researched were constructed to simulate urban combat and to serve as analytical tools for studies on combat in urban terrain. Although both were shelved in 1976, there are some portions of these models which can be extracted for use in the urban combat model. The inclusion or exclusion of these areas are discussed below:

1. MOBACS

The exclusion of the terrain model, discussed in A.1.b above, is based on the complexities of its construction, utilization and the related functions. The concept of the density contour map and the follow-on calculations for P(line of sight) and P(movement) based on this particular terrain model, is not appealing. From an explicit model of urban terrain, i.e., four walls, a roof and windows to

represent a building, line of sight or movement functions are either yes or no; there should not be a reason to calculate the probability that these functions occur. Therefore, the MOBACS terrain model is not further considered.

The one feature of the MOBACS model that is easily emplaced in the urban combat model, is the calculation of projectile penetration into a given structure. The equations and parameter constants from MOBACS are in the urban combat model but are not utilized. To fully employ these equations, research and computations must be done to calculate the velocity of the projectile as it enters the building and then determine incapacitation based on this reduced velocity and the location of the soldier. A follow-on to this module will be the completed formulas from AMSAA that will calculate the effects of fragmenting projectiles on buildings.

2. <u>URBWAR</u>

The terrain representation of an urban area in URBWAR is the desired method. The rectangular configuration of the buildings and the openings realistically describes a "typical" urban area and is amenable to explicitly determining lines of sight; the angular construction of the terrain makes the calculations of lines of sight and movement a geometric and trigonometric exercise.

A portion of the detection routine is similar to the one contained in the STAR combat model and is discussed in detail in Chapter III. The URBWAR model also contains,

although not stated explicitly, another detection method based on the glimpse model. The incorporation of this type of model as an addition to the STAR detection model is discussed in Chapter V; the methodology and the equations for the glimpse model are also discussed in Chapter V.

Because of the lack of dynamic movement in the URBWAR model, this module is not considered for use in the urban model. The ideas of which types of movement require modelling discussed in paragraph B3 should be considered. Chapter V discusses fully the movement logic and requirements for the urban combat model, with references to the URBWAR model.

III. OVERVIEW OF STAR

A. GENERAL

The models discussed in Chapter II, MOBACS and URBWAR, are the state of the art models for high resolution urban combat. They are both written in the FORTRAN IV computer language, a language not originally designed explicitly for event sequenced simulations. SIMSCRIPT II.5, on the other hand, is an event oriented language that is currently used in a medium-scale, high resolution combat model. This model is the Simulation of Tactical Alternative Responses, STAR, ground-air combat model, which is managed and operated at the Naval Postgraduate School by the TRADOC Research Element Monterey, TREM.

The initial incorporation of the urban combat model into the STAR model was to be a modular input. After initial research into the existing state of the art models, the STAR model and the initial formulation of the proposed urban combat model, this particular idea was suppressed. The urban combat model is now designed as a stand-alone model. However, the STAR combat model sequences the combat events in such a manner that the logic flows and event flows are easily followed and this same method of sequencing activities can be transformed into the urban combat model.

Section B of this chapter discusses the flow of events in the STAR model. Section C isolates the detection procedure

as modelled in STAR for dismounted soldiers. Section D discusses the terrain model and two of its related functions. Section E lists and discusses those events and routines from the STAR model that have been or should be incorporated in the urban combat model.

B. AN INTRODUCTION TO STAR

In June 1979, Lieutenant Colonel Edward P. Kelleher presented a paper to the 43rd Military Operations Research Symposium (MORS) at West Point, New York. The paper, untitled, is an introduction to the STAR combat model. The first portion discusses the history of the model and modifications to the model which are forthcoming. The next sections discuss the parametric terrain and the line of sight function; these particular aspects of the model are discussed in section D of this chapter.

The remainder of Section B of this chapter is an excerpt from the paper which discusses the flow of event in the simulation and the routines utilized within each event.

> Simscript is an event-oriented language, and STAR can best be understood in terms of the flow of events that take place in representing the combined arms battle. Figure 1 represents the basic event flow in STAR. Event STEP.TIME is the event that drives the rest of the simulation. The event updates the position of each element on the battlefield, and then checks line-of-sight between each pair of Blue and Red combatants. If LOS exists, the percent visible of the target is used to compute an "equivalent range" to a fully exposed M60 tank. That range is used, in conjunction with target crossing velocity, terrain complexity and a random number, to generate a "time to detect" for the observer-target pair.

If that time is thirty seconds (user input) or less, the next event in the chain, DETECT, is scheduled in the appropriate amount of time. The thirty second criteria is based on two considerations. First, in thirty seconds of a mechanized conflict many things change, affecting the detection rate, and eventually the calculated time to detect will no longer be valid. Second, the detection model in STAR is basically the DYNTACS detection model. That was based on field experiment data (using M60 tanks) and is only valid for times less than or equal to thirty seconds.

If the calculated time is thirty seconds (user input) or less, the next event in the chain, DETECT, is scheduled in the appropriate amount of time. The thirty second criteria is based on two considerations. First, in thirty seconds of a mechanized conflict many things change, affecting the detection rate, and eventually the calculated time to detect will no longer be valid. Second, the detection model in STAR is basically the DYNTACS detection model. That was based on field experiment data (using M60 tanks) and is only valid for times less than or equal to thirty seconds.

If the calculated time is greater than thirty seconds, the detector tries again at the next occurrence of STEP.TIME. At every STEP.TIME each element, Red or Blue, has an opportunity to schedule one or more detections.

When event STEP.TIME is completed, the simulation time is advanced until the next scheduled event, DETECT, for example. Two pieces of information are passed into DETECT: the detector and the entity to be detected. The program checks to see whether both entities are alive and whether LOS still exists. If these conditions and others are met the detection occurs. At this point the detected target is added to the list of targets belonging to the detecting entity, and event TARGET.SELECT is scheduled.

This event does just what its name implies. If the detector is not engaged in firing, he selects the "best" target on the list as a candidate for engagement. The selection is based on the amount of positive fire control being simulated for the selector's weapon type. Table 1 gives an example of several rules of engagement.

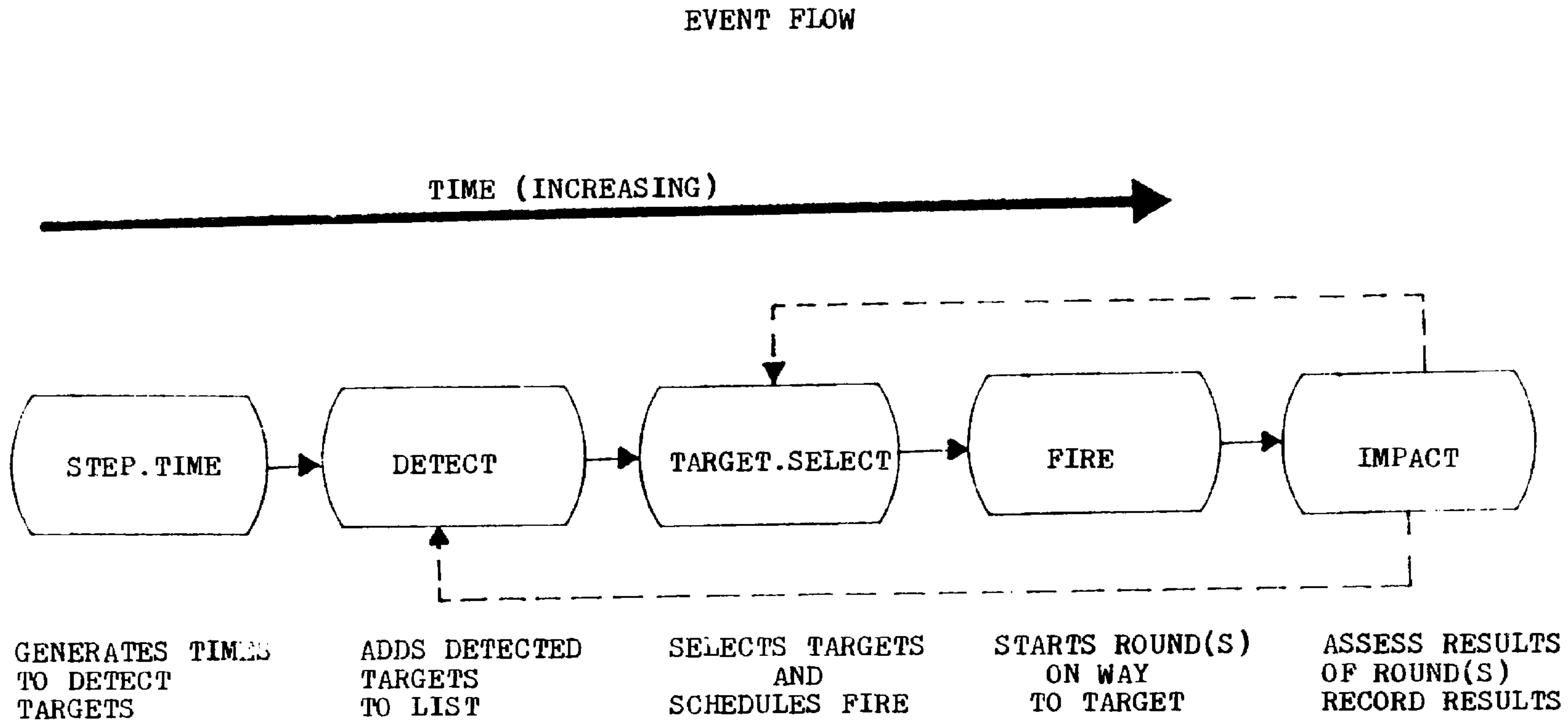


Figure 1. Event Flow in STAR

TABLE 1. Target Selection Tactics

RULES OF ENGAGEMENT

1. Platoon Leader Handoff
2. Highest Priority Target
3. Highest Priority Target Unengaged by Platoon Member, else Highest Priority Target
4. Highest Priority Target unengaged by a Company Member, else Highest Priority Target (engage with automatic weapon fire only).
5. Highest Priority Target Unengaged by a Platoon Member, else do not Engage

---

The first two rules in Table 1 are widely accepted as being used by various armed forces. Platoon Leader Handoff implies that the selector attempts to acquire the target his platoon leader is engaging. If he acquires that target, he engages it; if not, he does not fire. The second rule of engagement is that each entity always selects his own highest priority target. The other tactics are self-explanatory and reflect various levels of fire control. Figure 2 shows an example of the combined tactic: Platoon Leader Handoff, followed by highest priority not engaged by another platoon member. If all available targets are engaged, engage highest priority is with automatic weapon fire rather than with main gun or missile. In Figure 4, Blue 1 thru 4 are members of the same platoon. BLUE 1 is selecting a target. He attempts to acquire RED 2, who is being engaged by BLUE 3, the platoon leader. Lack of LOS prevents this. The two targets on Blue 1's list are both engaged by other platoon members. Blue 1 therefore selects his highest priority target, RED 3, but engages him with automatic weapons fire only. The selection of various levels of positive fire control and fire distribution methods gives the user substantial flexibility in investigating possible tactics.

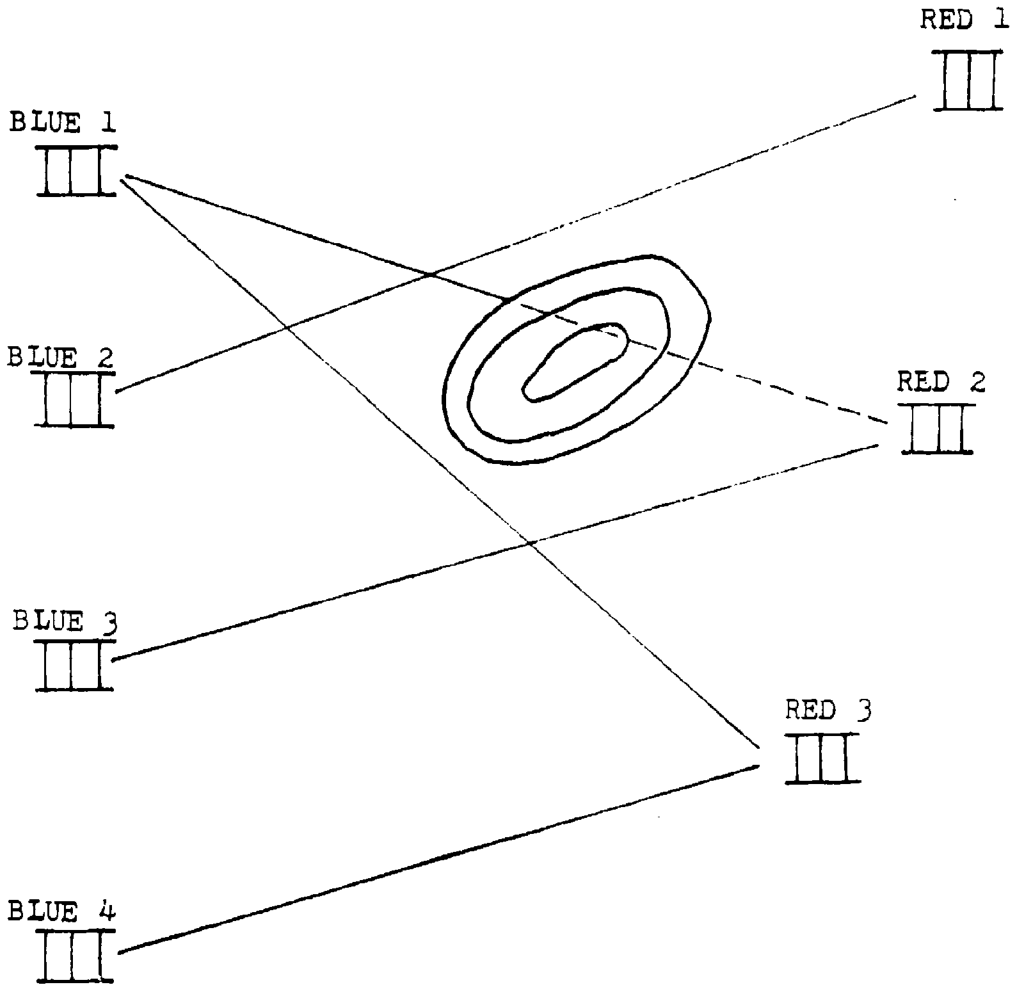


Figure 2. Sample Target Selection

When the target selection procedure is finished, if a target has been selected, event FIRE is scheduled at a time based upon a random draw from the user-specified lay and load time distribution. At the appropriate time the event FIRE is executed. Again line-of-sight is checked, as is the ALIVE.DEAD status of both the firer and the target. As will be discussed later, STAR differentiates between various types of damage and kill. A firer who has suffered a firepower kill or a catastrophic kill is not allowed to fire. Targets which have suffered a catastrophic kill and targets which have been mobility and firepower killed for at least a user-specified period of time are not fired upon. If all the necessary conditions are met, the FIRE takes place and event IMPACT is scheduled based on the range to the target and the velocity of the fired round. The identity of the firer and target, and the percent of the target that the firer could see at the moment of fire are passed to event IMPACT as arguments.

At the appropriate time, event IMPACT occurs. First LOS is checked. If no line-of-sight exists the round is assessed as a miss. If any part of the target is visible, the routines to determine whether or not a hit occurred are called, starting with COMPUTE.

Routine COMPUTE determines the parameters of the distributions of the miss distances in deflection and elevation for tank main guns, anti-tank guided missiles, and large caliber direct-fire weapons. These miss distances are assumed to be normally distributed. The parameters are biases and standard deviations. Their magnitudes are based upon a number of factors, including type and speed of the firer, speed of and range to the target, and whether or not the firer has fired at this target and sensed the preceding round. Having determined the biases and standard deviations, COMPUTE calls GEOM.

Routine GEOM draws from a normal distribution and converts the miss distance distribution parameters into a miss distance in elevation and a miss distance in deflection. Using the percent of the target visible (as opposed to uncovered) at fire, the firer's aim point is then computed. If the "turret ring" of the target was visible, the firer is assumed to have selected the turret ring as his aim point. If the turret ring was concealed at FIRE, the firer is assumed to have selected the vertical center of the visible part of the target as his aim point. In either case the firer is assumed to have selected the horizontal center line for his aim point. Errors in this selection are at least partially included in aim point bias parameters.

Having determined the aim point, GEOM assesses whether or not the round missed in elevation. This occurs in three ways. First: the elevation miss distance is positive and larger than the distance from the aim point to the top of the turret. Second: the elevation miss distance is negative and larger in magnitude than the uncovered portion of the target below the aim point. The third way in which a miss in elevation can happen is only possible in the event that the selected aim point (based on amount visible, i.e. unconcealed at FIRE) is now covered. Figure 3 shows this case. When this occurs, the elevation miss distance may be positive and still result in a hit on the covered portion of the target, i.e. a miss. The second and third cases result in a "sensed" miss. The first case and all misses in deflection result in an "unsensed" miss. If the round did not miss in elevation, deflection is checked. First the aspect angle between the fore and aft center line of the target and the line of sight from the firer to the target is determined. This angle and the target horizontal dimension are used to determine the effective width of the target from the point of view of the firer. Figure 4 shows three possibilities. It should be noted that the size and direction of the elevation miss are used to determine whether the horizontal measurements used should be those of the turret or those of the hull. When the effective width has been computed the determination of hit or miss in deflection is made. Since, by assumption, the aim point is always the center of the target, if the deflection error is larger than half the effective width of the target, the round missed in deflection. If at any point it is determined that a miss occurred, control is returned to event IMPACT, with an indication that the round missed. If the round did not miss in elevation or deflection, a hit must have occurred. In that case, GEOM determines (by table lookup) the mobility damage function, firepower damage function, mobility and firepower damage function and probability of catastrophic kill. These are all numbers between 0 and 1, (representing probabilities). The size of the number is a function of the type of projectile fired, the type of target, aspect angle, dispersion class and target exposure. Dispersion class is used to account for the fact that a round with a large dispersion is less likely to cause damage than a round with a small dispersion. Target exposure accounts for the difference in vulnerability between a target in hull defilade and one fully exposed. In STAR a target with more than fifty percent of the hull height covered is in "hull defilade" for damage assessment; all others are fully exposed.

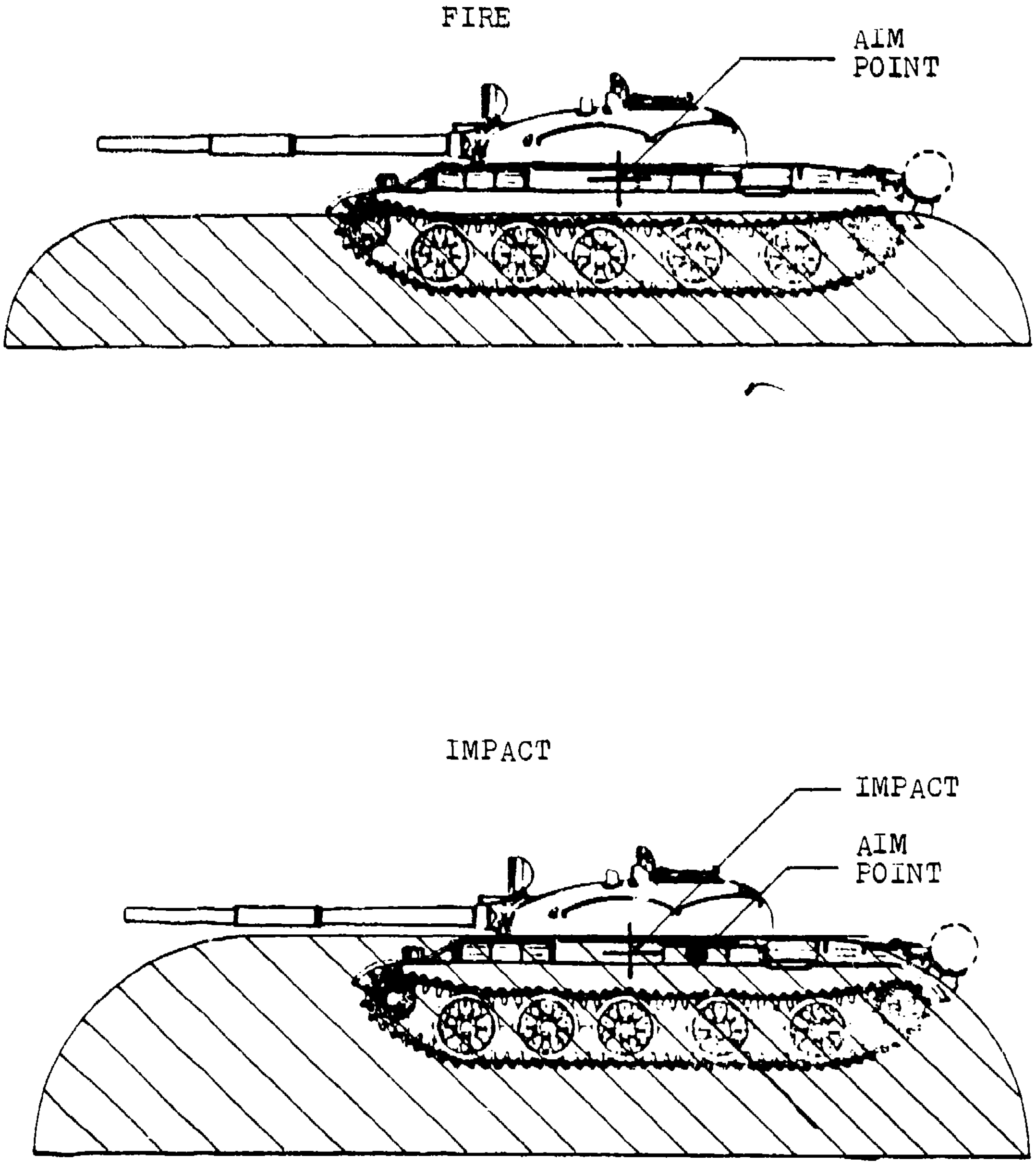
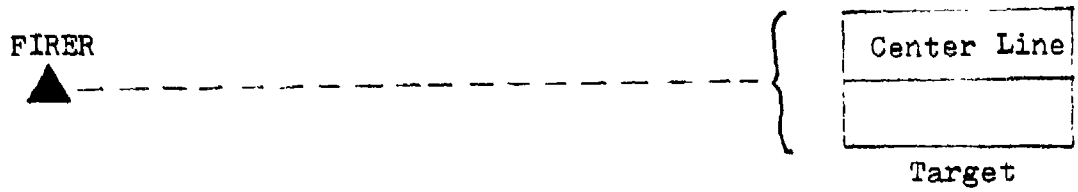
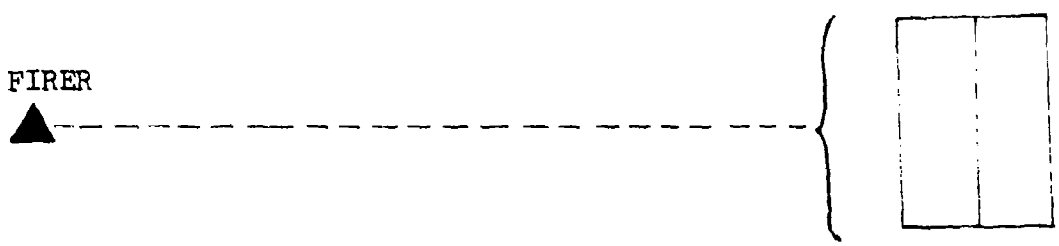


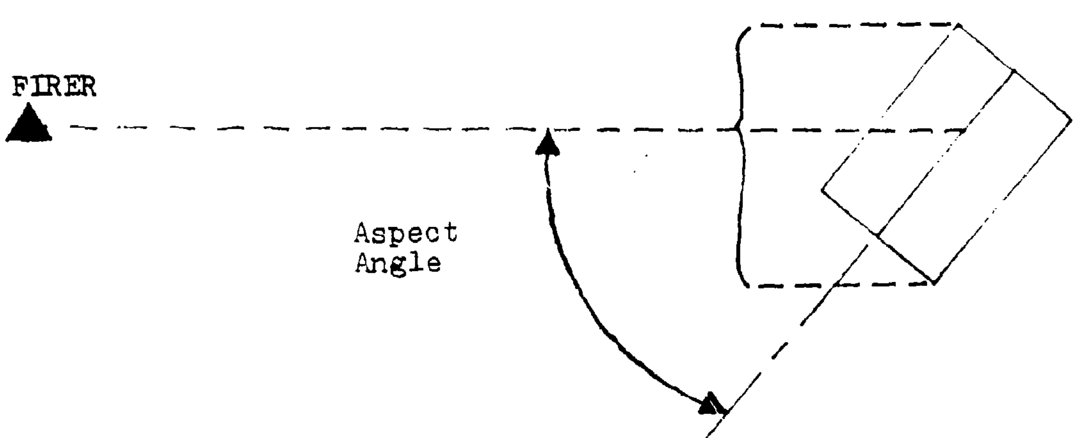
Figure 3. Aim Point Covered



Effective width is front horizontal dimension



Effective width is side horizontal dimension



Effective width=f(angle, side horizontal width, front horizontal width)

Figure 4. Apparent Width of Target

When the damage functions and probability of catastrophic kill have been determined, GEOM calls routine ATRIT. It should be noted that routine GEOM, with its use of miss distance to determine whether or not a hit occurred, is not used for automatic weapons. No bias and standard deviation data were available for those weapons when STAR was coded. For automatic weapons, COMPUTE calls routine SUBCAL, which uses probability of hit and the assumption of round-to-round independence, to determine the number of hits. Based on the number of hits, the damage functions and probability of catastrophic kill are determined and passed to ATRIT.

Routine ATRIT uses the damage functions and probability of catastrophic kill to determine the outcome of the engagement.

STAR distinguishes among four different types of kills; mobility (M kill), firepower (F kill), mobility and firepower (M and F kill), and catastrophic (K kill). A catastrophic kill is immediately apparent to all participants in the battle. Other types of kills are not immediately apparent. A mobility-killed entity stops moving. A firepower-killed entity is incapable of firing any armament, but may still move. An entity which is mobility and firepower killed may neither move nor fire.

Entities which are K killed are not fired upon again. Rounds already fired at the entity are allowed to impact, but no new rounds are fired. Unguided rounds which the K-killed entity had already fired are unaffected, but command-guided rounds such as TOWs, Dragons and Saggers are assessed as misses.

M kills and F kills are not sufficient cause for ending engagement and do not affect the selection of the entity as a target. M and F kills are also not immediately apparent and the entity may be selected as a target for a period of time (which is a user input). At the end of that time it becomes apparent that this entity is no longer involved in the battle and the entity is not eligible for future target selections.

The results of the engagement are determined by drawing a random number from a uniform distribution interval 0 to 1. This one number is then used for all comparisons for the engagement. First, the probability of catastrophic kill is compared to the random number. The catastrophic kill phenomenon is assumed to be independent from round-to-round. If the random number is smaller, a catastrophic kill is assessed. Otherwise, routine ATRIT uses the same random number to determine whether any other type of kill occurred.

It is intuitively appealing to believe that an entity is easier to M kill or F kill after it has been hit several times than on the first hit. The Ballistics Research Laboratory (BRL) supplied mobility and firepower damage functions model that phenomenon. They are not, in general, directly probabilities. They are measures (as we understand it) of the proportion of the entity's remaining capability that it loses as a result of this hit. Each entity has three attributes which are used to keep track of accrued damage. They are: M.D. mobility damage; F.D., firepower damage; and MF.D, mobility and firepower damage.

The user determines, by input of a variable LALPH (lethality alpha), how much of the previously incurred damage is to be considered in assessing the results of this round. LALPH is a number between zero and one. If LALPH is set to zero, none of the previously incurred damage is considered, and the probability of each type of kill is independent from round to round. If LALPH is set to one all previously incurred damage is considered. In this case a vehicle which has been hit several times is much more vulnerable than one which has never been hit. In particular, as LALPH is increased, for any hit on a vehicle except the first one, the vehicle is more apt to suffer a non-catastrophic kill than if LALPH were smaller, and a larger share of the probability is shifted into the M and F category.

To determine whether a non-catastrophic kill has occurred (given no catastrophic kill), the current levels of M damage, F damage and M-and-F damage this impact. These levels are then compared to the previously drawn random number. Four outcomes are possible: M-Kill on this shot, F-Kill on this shot, MF-Kill on this shot, or no kill on this shot. If the target has been MF-Killed on this shot, or both M-Killed and F-Killed (as a result of this shot and some previous engagement), a FINAL.DEATH is scheduled to occur in the user input amount of time (T.LINGER). Table 2 shows the results of increasing LALPH from 0 to 1 for a specific set of damage probabilities.

No degradation of the target's ability to move and fire is assessed for levels of M or F damage which do not result in a kill.

Before control is returned to event IMPACT, the global variable DAMAGE.NUM is set to a value which indicates the results of the round. Table IV presents those values and their meanings.

TABLE 2. P(k) as a Function of LALPH

SITUATION: Tank #47 has been hit by a round. Before the impact, #47 had incurred the following damage M.D. = .55, F.D. = .65, MP.D = .75 (Note that these numbers are Union i.e., M.D. is the union M-only and M intersect F).

DAMAGE FUNCTION FOR THIS SHOT:

M = .7, F = .6, MF = .8, P(K) = .1

FOR LALPH = 0 (INDEPENDENT, ROUND TO ROUND)

P (MK) = .2 (M F - (Only F (M F)))

P (FK) = .1 P (MFK) = .4

FOR LALPH = 1.0 (ALL PREVIOUS DAMAGE CARRIED FORWARD)

M.D. = (1.0) * (.55) + (1.0 - (1.0) * (.55)) *.7 = .865

F.D. = (1.0) * (.65) + (1.0 - (1.0) * (.65)) *.6 = .860

MF.D = (1.0) * (.75) + (1.0 - (1.0) * (.75)) *.8 = .95

P (MK) = .95 - .86 = .09, P(FK) = .085, P (MFK) = .675

PROBABILITIES OF KILL on this round, given LALPH

	LALPH =0	=0.05	=0.5	=0.75	=1.0
M	0.2	0.1945	0.145	0.1175	0.09
F	0.1	0.09925	0.0925	0.08875	0.085
MF	0.4	0.41375	0.5375	0.60625	0.675
K	0.1	0.1	0.1	0.1	0.1

Upon return to event IMPACT, the results of the engagement are examined, and the firer takes action as directed by the tactical routines. He may re-engage the same target, select a new target, or move to a new firing position.

C. DETECTION OF DISMOUNTED INFANTRY IN STAR

1. Background

During the simulation the event DETECT is scheduled based on a time to detect that is returned from routines CARDIO and DISTMD.CARDIO. As stated in LTC Kelleher's paper, the detection routines in the STAR combat model were based on the DYNTACS detection model. An enhancement to the DYNTACS model was the ASARS version which modelled detection for dismounted infantry. This model in turn was modified by Captains Howard J. Carpenter and Edward E. Thurman in [Ref. 1]. Their revision is incorporated in the STAR combat model. This routine is called DISMTD.CARDIO.

The continuous detection model in DYNTACS computes a probability of detection during t seconds. The computational form is:

P = 1.0 - exp**(-L*t)

where L is the detection rate function (assumed constant) and t is the amount of time (in seconds). This value P is drawn from the population distribution of the detection times and is used to compute:

t = ln(1.0 - P) / (-L),

with detection times occuring in clusters of t second intervals.

The constant L is empirically calculated by an equation derived from test results. This test was conducted at Fort Knox, Kentucky in 1965, not as a search experiment, but rather as a detection experiment. Various scenes were filmed of an M60 tank against various types of backgrounds on terrain similar to Western Europe. The maximum range to the target was 1500 meters. The film was shown to the test subjects, who were stationary and viewing with the unaided eye. The test data was collected and the following formula resulted:

L = P(k) * (- .003 + (1.088 / d)) with d defined as:

d = 1.453 + tc * (.05978 + 2.188 * (R**2) - .05038 * cv)

The variables are defined as follows:

a. tc - a terrain complexity factor. The more complex the terrain the higher the number. The range is from one to seven and is related to the number of potential avenues of approach.

b. R - the apparent range to the target in kilometers. This value is based on the height of the target not the visible area. All heights are scaled against the height of an M60 tank. For example, an M113 Armored Personnel Carrier is .66 the height of an M60, therefore

R(M113) = R(M60) / .66

c. cv - the crossing velocity of the target, i.e. the component of the target's velocity vector perpendicular to the observer-target line of sight in meters per second.

d. P(k) - the probability the observer is looking in the sector of the target. To calculate P(k), the modelling approach is to divide the potential search sector into twelve subsectors of thirty degrees each. Each subsector is assigned a probability. (Figure 5) these probabilities represent the probability the observer is looking in the sector of the target. For example if the observer's primary direction of search is indicated as Y and the target is located at point A, the probability the observer is looking at the target is .14 = P(k).

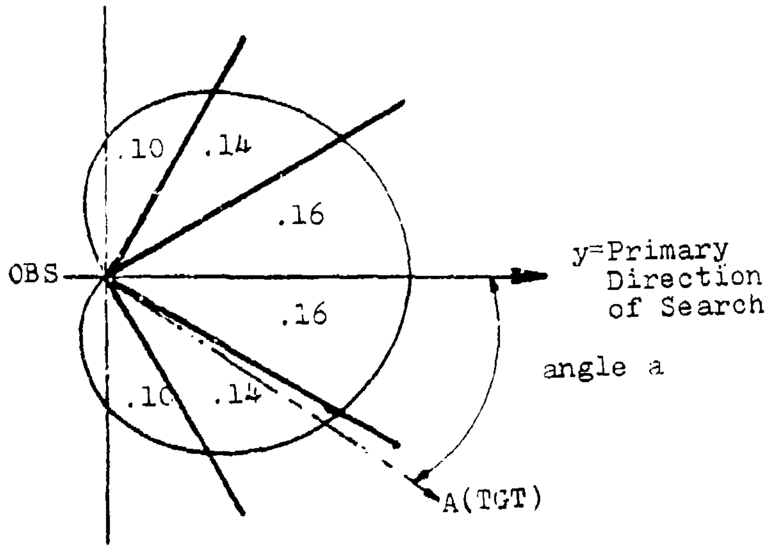


Figure 5. Cardioid Used in the Detection Models

The ASARS detection model modified the DYNTACS model to account for the abilities of the dismounted soldier. Additionally, the factor of aural cues to the location of potential targets was added, but this is not modelled in STAR, nor will this modelling technique be discussed.

Reference 6, Volume IIB, has the source code and discussion on this topic. The portion of this detection model, assuming perfect intelligence of the target, is incorporated in DISMTD.CARDIO.

2. Routine DISMTD.CARDIO

DISMTD.CARDIO has reduced the soldier's search sector to be initially ninety degrees. The value is dynamically changed based on the soldier's status in the simulation. All potential values of the soldier's search sector are expressed in multiples of thirty degrees. If he has no detections this value is set to 360 degrees and the DYNTACS cardio is used; if he has detected a target the area is reduced to thirty degrees. An adjustment to the cardoid used in DYNTACS is made to account for this reduction in the search sector. (Figure 5) The observer whose search sector is less than or equal to ninety degrees has at least a fifty percent chance of detecting the target within the sector assuming the angle:

a = (obs-tgt angle) - (obs primary direction of search)

is within this ninety degree sector.

From [Ref. 1], the search sector of the observer is a variable called AREA. The variable P.SUB.K is set to 30.0/AREA if the AREA is not equal to 360. If AREA equals 360, the following formula is used:

P.SUB.K = 1/12 + (1/2pi) * (sin(a + pi/6) - sin(a))

P.SUB.K is Monte Carlo'd against a uniform (0,1) and a detection event will be scheduled if P.SUB.K is less than or equal to the random number. The detection event is scheduled at a time increment from the current clock time. The time increment is distributed log-normal and the formula to calculate this time is taken directly from the ASARS detection model.

If the random variable Y is distributed log-normal, the natural log of Y is distributed normal. Therefore, Let:

a. NDT = ln (Y) such that NDT is distributed N(u,s), with u = E(NDT) and s = SD(NDT).

b. Draw a random Normal (u,s) random number.

c. Set NDT equal to the random number and

d. Calculate the increment in time for detection to be:

Y = detection time increment = e ** NDT

The log-normal random variable does not have the memoryless property as the exponential analog to the glimpse model (discussed in Chapter V). This particular property of the log-normal random variable implies that the longer you have been searching, without a detection, the less likely you will find the target in the near future. In symbols,

$$P(Y < 1.5 \mid Y > 1) < P(Y < .5)$$

This theory of detection is intuitively appealing and has been used as the model for personnel viewing a dismounted target at 400 to 500 meters with the unaided eye.

To compute the mean of the random variable NDT, the ASARS model introduced three variables: the range, crossing velocity and terrain effects. These three variables are all used in the computation of this mean time to detection. From data collected in the Target Acquisition Study (TAS), a visual detection algorithm was developed for the dismounted infantry and the range and crossing velocity effects were defined as follows:

$$G \text{ (range effect)} = .018 + .0058 * R$$

$$B \text{ (velocity effect)} = 1.39 - .076 * cv$$

The terrain effect, A, is handled by a direct assignment depending on the type of vegetation in which the target is located. [Ref. 6; p. IV-A-17; Vol. IIA] The variable A can assume any of three values: 0.53, 1.49 and 1.35. [Ref. 1; p. 158] The formula which links all three effects to the mean time to detection is:

$$E(NDT) = -2.2 + A + B + G$$

and the standard deviation of NDT is 0.81. The value of the standard deviation is based on the mean square error as found in an analysis of variance. Appendix B of [Ref. 6; Vol. IIA] contains a detailed description of the development of this model, to include the experimental design to conduct the analysis of variance.

The TAS data did not take into account the differences in urban terrain. As stated in the URBWAR section,

Chapter II, it is not apparent that this routine will return realistic detection times without the benefit of validated test data on urban terrain. Another approach, which may be incorporated in the urban model, is the Glimpse Detection Model discussed in Chapter V.

D. PARAMETRIC TERRAIN, LINE OF SIGHT AND MOVEMENT

1. Parametric Terrain

In the years preceding 1976, the accepted approach in high resolution combat simulations was to represent the macro terrain (defined as the computed value of the elevation for any (x,y) pair [Ref. 2; p. 5]) as digitized terrain. The terrain was stored in a large table of numbers representing (x,y,z) coordinates in the Cartesian system. The procedure is to input an (x,y) pair and the matching z-coordinate is found through a table lookup algorithm. If one wanted to approximate the smoothing of terrain, because this terrain representation resembles a field of spikes, interpolation between points was a computationally simple procedure. But like anything associated with ease of computations, there is a price to pay - that cost being storage requirements for the (x,y,z) combinations. As an example of this cost, to store the elevations of a 30 kilometer by 60 kilometer battlefield at 100 meter intervals requires a storage capacity of 180,000 words (720,000 bytes) [Ref. 1; p. 53]. Because of these constraints most battlefields for Battalion models have been restricted to a 10 kilometer by 10 kilometer box [Ref. 2; p. 5].

In 1976, MAJ Chris Needels proposed a method of representing terrain through a parametric procedure. The concept was to model individual hill masses and then form the terrain by overlaying these hill masses. In his model, MAJ Needels represented analytically the hill mass as a scaled bivariate normal probability density function. The result being a bell-shaped cross-section and elliptical contours for each hill mass. By varying the parameters, all types of hill masses could be represented. The hill mass functions and ideas of MAJ Needels were generalized into the STAR parametric terrain model, which has significantly changed the computations of these functions by making them more efficient and easier to fit the real terrain maps [Ref. 2; p. 9].

A detailed description of input parameters, equations, computer language, etc. will not be presented in this section, but only those characteristics of the terrain and how the STAR combat model uses this terrain representation to perform two basic functions, line of sight (LOS) and movement. The detailed methodology for terrain processing is contained in [Ref. 2].

Depending on the terrain to be represented, in a typical 10km by 10km box of 100 hills, about 800 parameters (eight per hill) are required to be stored; comparatively, in digitized terrain slightly more than 10,000 numbers must be stored. This twelve-factor savings in storage is substantial [Ref. 2; p. 23].

The actual parameter collection process for each hill mass is done manually. These parameters are:

a. XC.H - x-coordinate of the hill center.

b. YC.H - y-coordinate of the hill center.

c. PEAK.H - elevation of the hill mass measured from sea level.

d. ANG.H - orientation of the ellipse measured in degrees counter-clockwise from the east to the major axis of the ellipse.

e. ECC.H - eccentricity of the ellipse (ratio of the major axis to the minor axis).

f. SPRD.H - measure of the hill size which is the distance in meters measured along the major axis from the hill center to the contour line 50 meters down from the peak.

g. HT.H - maximum height of the normal curve describing the hill mass; must be greater than 50 meters.

h. CUT.H - vertical distance measured down from the peak beyond which the hill mass is no longer considered in the computation. [Ref. 2; pp. 9-14]

It has been demonstrated that a person trained in the skill of parameterizing can complete a 10km by 10km box in five to seven work days (35 to 56 hours) [Ref. 1; pp. 58-50]; the result being the data input for the terrain model.

The STAR terrain model takes these eight basic parameters, and other inputs and creates the terrain

through a terrain preprocessor. The preprocessor develops and processes only those hills relevant to the particular area of interest. Thus, the speed and efficiency of performing functions on the terrain, e.g. line of sight, is optimized.

Forests are also modelled in the STAR combat model; the forests being approximated as fitted elliptical cylinders. The process of the tree preprocessor is analogous to the hill preprocessor, the result being a list of tree ellipses in the area of interest. Of particular note, is the inability of two combatants in the same forest ellipse, no matter what the distance between them, to be able to see one another. The logic currently in the program does not allow one to see into, within or out of a forest ellipse. This shortcoming is of particular interest because this function must be available if one is to model urban terrain using the STAR terrain model.

The research of the three combat models, MOBACS, URBWAR and STAR revealed certain characterisitics of urban terrain that are desired if one is to model combat in this environment. The rectangular structure of buildings is a necessary characterisitic as modelled in URBWAR. The rejection of the MOBACS terrain has already been discussed in Chapter II. The STAR parametric terrain is definitely not conducive for urban terrain; but in order to understand the functions performed using the

terrain, one must first understand the terrain model in STAR. Chapter IV will go into the final representation of the terrain in the urban combat model.

What was sought in investigating the STAR terrain model was its use in order to conduct the line of sight and movement functions. Although these functions were performed on terrain unlike the urban terrain modelled, the methodology which models these functions can be placed in an urban model.

2. Line of Sight (LOS)

The STAR combat model calculates the line of sight using the parametric terrain. Given an observer and a target, the line of sight routine calculates the fraction of the target visible to the observer, i.e. a number between 0 and 1. The fraction visible is based on the vertical height and not on area visible. If this model were to be used in urban terrain, a possible shortcoming in this procedure is apparent, e.g. a man on the tenth floor could see a prone man easier than a standing man because of the amount of area exposed as opposed to a purely vertical height. Through a series of checks and calculations, only those hills and forests that affect the line of sight are checked, the fraction visible is returned and the line of sight check is completed.

This simplistic explanation of this process does not adequately explain the full computational process that goes into line of sight checks. An interesting feature

of the STAR terrain model is the single parameterization of the line of sight, i.e. all line of sight checks can be made with a single dimensional parameterization of the line of sight line, normally described in three dimensions. The mathematics of the procedure are explained in [Ref. 2] and the utilization of this logic in the urban terrain model is in Appendix C.

3. Movement

Movement in STAR is controlled by user inputs into the model. These parameters describe the start point, intermediate points where turns must be made (a line segment approximation of movement across terrain) and an end-point. All routes are expressed in terms of the number of nodes (start, turning and end) in (x,y) pairs on the terrain. The rate of movement is calculated as a function of the weapon type, system type and slope of the intervening terrain along the desired route [Ref. 1; p. 134]. The intricacies of table look-ups, e.g. the max/min velocity and acceleration for a system, the overlaying of the terrain along the route to determine slopes and the subsequent movement of the troops can be incorporated in an urban model. This is discussed in Chapter V as future enhancement.

Of particular concern in the movement logic in STAR is the explicit calculation of movement rates based on tactical formations, missions, vehicles with or without soldiers, etc. The resolution of the macro terrain allows for this type of calculation; however, when one begins to

think of how to model a soldier moving through a building, the required higher resolution of the terrain representation has to be considered. This resolution consideration is discussed in Chapter V.

E. ROUTINES IN STAR ADAPTABLE TO URBAN COMBAT

The STAR combat model contains a myriad of routines and events that can be used in the urban model. These modules will have to be reformatted and adjusted to handle the proposed urban model, but the logic and thought processes of combat actions and assessments are similar whether they be within or external to urban terrain. The urban combat model contains portions of some of the routines in the STAR model and similarities between the two can be seen.

Some of the routines and events that can be placed in the urban model are:

1. STEP.TIME: A facsimile of this driver event is currently in the urban combat model, but is not in the configuration of the STAR routine. It is used only to schedule lines of sight for each observer-target pair.

2. DISMTD.CARDIO: Like STEP.TIME, this routine in the current urban model is a place holder. The times to detect that are returned bear no resemblance to the desired detection times.

3. TARGET.SELECT: This event has not been written nor has the target priority criteria been established.

4. FIRE: At present, this event is called KILL.MAN in the current urban model. The event randomly selects a probability of kill and based on the range to the target, the firer either hits or he misses. The inclusion of all events and routines associated with firing against a dismounted target as contained in the STAR FIRE event, i.e. IMPACT, LOS, INF.COMPUTE, ATRIT, INCAPACITATE, and BRST.FIRE, are all discussed in [Ref. 1]. For engagements against targets other than dismounted infantry, the discussion in LTC Kelleher's document covers all routines and events that need to be incorporated.

The adding of these routines and events is not an overnight task, nor is it to be construed as an easily defined task. Research into these routines and events and their subsequent incorporation into the urban combat model may require renaming variables, adding attributes, arrays and data requirements, and in perhaps some cases rewriting the event or routine keeping only the logic flow as a guideline.

Chapter V contains additional discussion and detail on using STAR modules as an enhancement to the urban combat model.

IV. GENERAL METHODOLOGY OF THE URBAN COMBAT MODEL

A. DESCRIPTION OF THE MODEL AND ITS CAPABILITIES

The urban combat model is structured to simulate a Red force versus a Blue force of dismounted infantry in typical urban terrain. The size of the forces to be simulated is not model dependent, nor is the size of the urban terrain to be used in the simulation. All combat units, regardless of the force size, are resolved down to the infantry soldier. The current version of the model is not capable of full combat simulation. The current capabilities of the model are:

1. Creates the urban terrain,

2. Creates the individual soldier,

3. Conducts lines of sight between observer-target pairs and returns the range to the target and the fraction visible of the target to the observer if there is line of sight. (These two values are to be used in routine DISMTD.CARDIO, discussed in Chapter III, which returns a time to detect.)

4. Places detected targets on the observer's target list and, if line of sight is lost, removes the target from the observer's target list, and

5. Conducts a pseudo-simulation, i.e. events have been created to schedule lines of sight, detections, firings,

and the assessing of casualties. These events were placed in the model, not as a final product, but rather as an aid to observe the functioning of the above four capabilities.

B. DESCRIPTION OF ENTITIES AND LINE OF SIGHT ARRAYS

The urban combat model defines two classes of temporary entities:

HOUSE - a building in the urban terrain and

MAN - an individual soldier.

The simulation creates a set called TOWN, which contains all HOUSES. Each HOUSE owns a set called BLDG.LIST in which are filed all MAN entities who are in that building.

1. Entity HOUSE

The temporary entity, HOUSE, is modelled as a rectangular, box-like structure. The user is required to design the composition of these entities based on his desires for the type of urban terrain to be utilized in the simulation.

The attributes of the HOUSE and an example of their utilization are listed below:

a. BLOCK: The number of the block in which the house is located. In the search algorithms, discussed in Section E of this chapter, the isolation of the houses in a certain block increases the efficiency of the algorithm.

b. DIM.FLOOR: The distance between floors in the house. The model currently assumes the house has equal spacing between floors. When calculating the attributes of

the combatants, a soldier can be placed on a floor of the house by a division of this value into his z-coordinate.

c. DIM.X,DIM.Y,DIM.Z: Dimensions of the building in the x, y and z directions respectively. This model assumes regularly shaped, box-like structures. These values are used extensively in the line of sight routines in the testing of the building for the line of sight checks.

d. KM: The constant associated with the structure as described in the MOBACS combat model as k(m), Chapter II. Although not utilized in the current urban combat model, it will be used to assess non-fragmenting projectile penetration of a building.

e. NAME: Denotes the number of the building. This number relates to the attribute of the combatant, MANS.HOUSE, given that he is in the HOUSE. It is additionally provided as a means to count through the houses when doing the line of sight checks.

f. N.OPENINGS: The number of openings, defined as windows or doors, in the house. These openings are only those that face to the outside of the building. It is assumed that movement and line of sight within the building, given two combatants are on the same floor, is not hindered by a lack of openings to move or see through. This value is linked to the three opening arrays, OPENING.ARRAY, OPEN.DIMEN and OPEN,FACE, which contain all the information concerning a specific opening. These arrays are more fully explained later in this section.

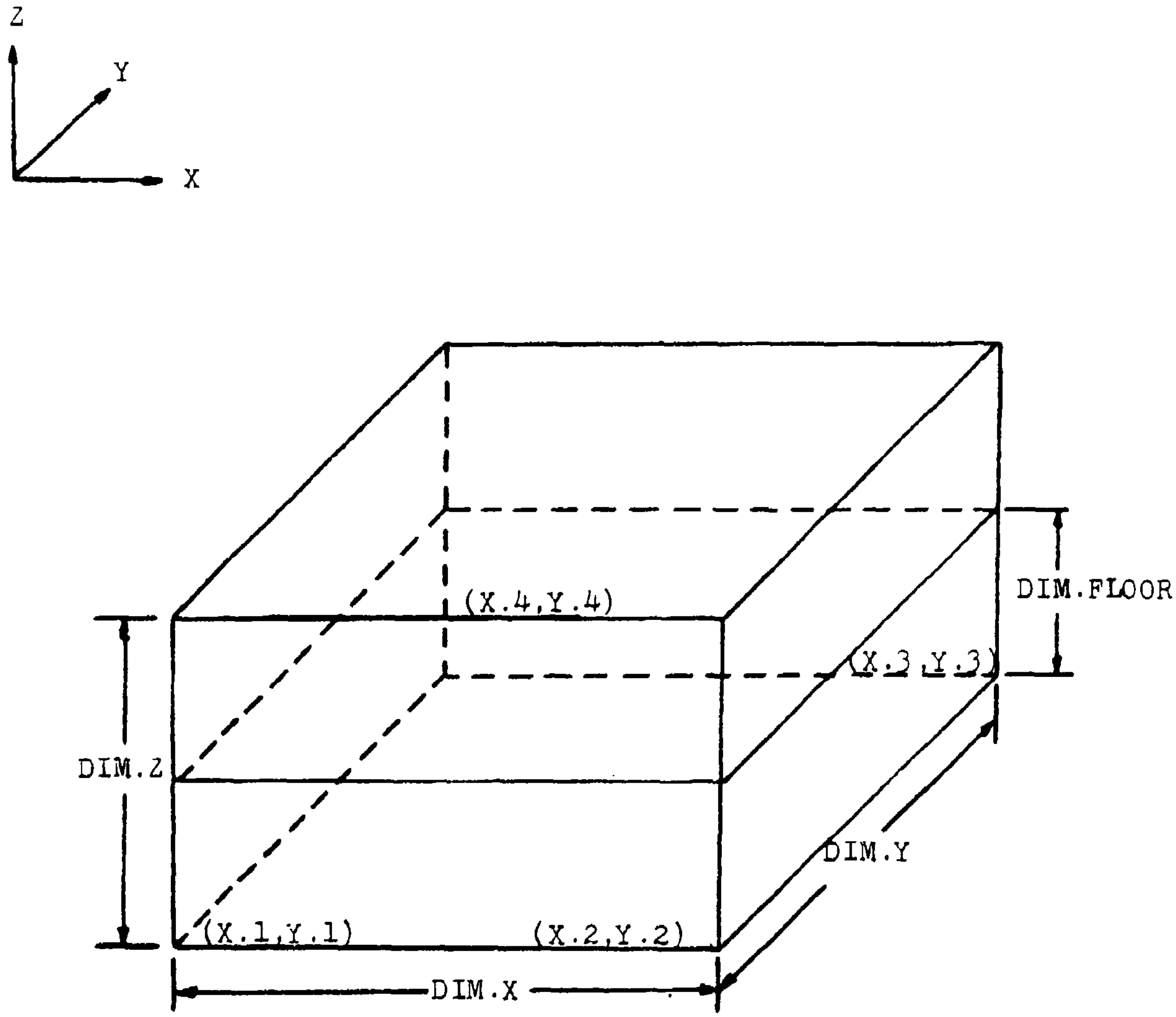
g. X.1,X.2,X.3,X.4: The x-coordinates of the building. They are numbered in sequence from the bottom left (southwest) counter-clockwise around the building.

h. Y.1,Y.2,Y.3,Y.4: Analogous to the x-coordinates.

i. THICK: Value for the thickness of the structure in inches. This value is later changed to be expressed in meters and will be used in the calculation of non-fragmenting projectile penetration as described in MOBACS. Each building is considered homogeneous in thickness.

j. TYPE: Denotes the type of structural material used in the building. A building can be of only one type. The assignment of the KM attribute is directly related to the type of building material used in the building.

All attributes of the buildings associated with the cartesian coordinate system are input into the program as offsets from a known point and are depicted in Figure 6. For example, the (X.1,Y.1) coordinate pair describe the lower left hand coordinate of the building, as you face the building, as an offset from the origin of the grid designation (0.0,0.0). All other coordinates of the building in the X-Y plane are then calculated from (X.1,Y.1) based on the dimensions of the structure which are input values. A single dimension is input for the elevation of the building. The model assumes that all buildings are sitting on tabletop terrain, i.e. elevation zero, therefore only one input is required for the Z-coordinate of the building.



INPUT DATA REQUIREMENTS

(X.1,Y.1)
DIM.X
DIM.Y
DIM.Z
DIM.FLOOR

Figure 6. Building Representation in the Urban Combat Model

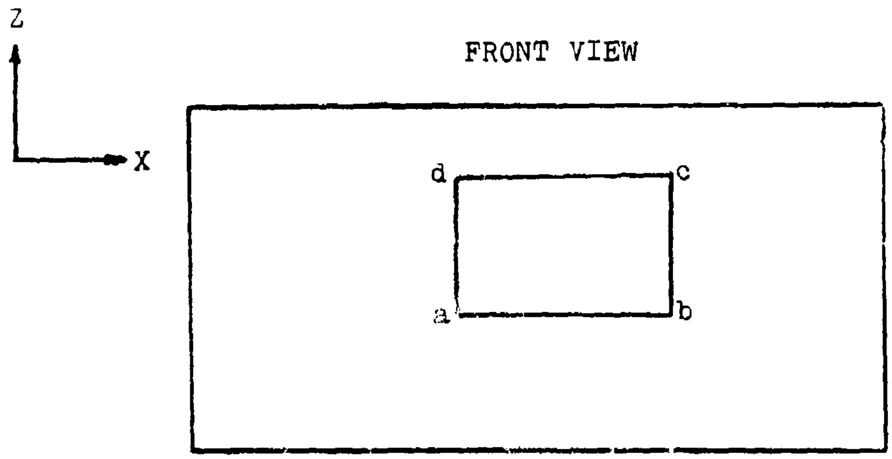
Additionally, the coordinates of the openings are input as offsets from (X.1,Y.1) and describe the lower left hand corner of the opening as you face the opening from the outside. (Figure 7) There are three arrays designated to contain the values to describe the openings:

a. OPENING.ARRAY: This array is dimensioned as the number of houses by 3 * the number of openings in the HOUSE. It contains, in groups of three, the x, y and z offsets of each opening in the building from the (X.1,Y.1) coordinates of the building. (Since the base elevations of all buildings are at zero elevation, the z-offset is merely the elevation of the opening.)

b. OPEN.DIMEN: The dimensions of this array are the same as the above array. The values are input in groups of three and are the dimensions of the opening in the x, y and z directions.

c. OPEN.FACE: The array is dimensioned as number of houses by number of openings. The values it can contain are integers from one to four, corresponding to south, east, north and west, respectively, indicating the direction in which the opening is facing.

All calculations which involve the opening are done from the base coordinate of the opening and the dimensions of the opening are used accordingly.



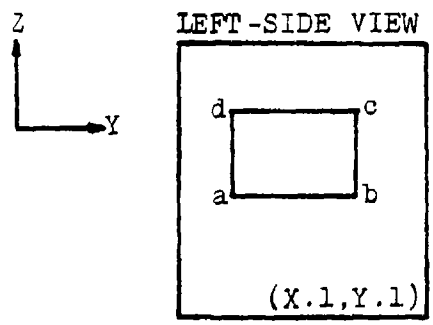
a = (X.1 + OPENING ARRAY (I,1),
Y.1, OPENING ARRAY (I,3))

b = (a_X + OPEN.DIM (I,1), a_Y, a_Z)

c = (b_X, b_Y, b_Z + OPEN.DIM (I,3))

d = (a_X, a_Y, a_Z + OPEN.DIM (I,3))

NOTE: (1) OPENING ARRAY (I,2) = 0.0
(2) OPEN.DIMEN (I,2) = 0.0
(3) This is the first opening in HOUSE (I)



a = (X.1, Y.1 + OPENING ARRAY (I,5),
OPENING ARRAY (I,6)

b = (a_X, a_Y + OPEN.DIMEN (I,5), a_Z)

c = (b_X, b_Y, b_Z + OPEN.DIMEN (I,6))

d = (a_X, a_Y, a_Z + OPEN.DIMEN (I,6))

NOTE: (1) OPENING.ARRAY (I,4) = 0.0
(2) OPEN.DIMEN (I,4) = 0.0
(3) This is the second opening in HOUSE (I)

Figure 7. Opening Representation in the Urban Combat Model

2. Entity MAN

Each MAN is modelled as a temporary entity. His list of attributes is not as entensive as in the dismounted infantry version of STAR, but was constructed only to allow the capabilities described above to be exercised and verified. As the model matures, more attributes can be added.

The attributes of the MAN are:

a. ACTIVITY: Denotes the type of activity in which the combatant is currently engaged. The type of function the combatant can perform is directly related to his activity status, e.g. if a combatant's status is dead, he can perform no functions or if a combatant is firing, he cannot detect targets until his engagement event is over.

b. ARMY: The value that links the combatant to a force. This value assists in the control of who can detect or engage whom, i.e. a Red combatant will not engage or detect another Red combatant.

c. MANS.BLOCK: Denotes the block (grid square) in which the combatant is located. When determining the attributes of the soldier, only those buildings in the soldier's block will be checked, thus increasing the efficiency of the assignment algorithm.

d. MANS.DIR: This value indicates the direction in which the combatant is looking. The correlation of the integer value to the direction is analogous to the elements of the OPEN.FACE array. When conducting line of sight

checks. this value directs the flow of the program to the correct subroutine. This is explained further in the discussion on the line of sight procedure.

e. MANS.FLOOR: The value of the particular floor on which the combatant is situated. This value is used in the line of sight checks as follows: if two opposing soldiers are on the same floor of the same house, line of sight is yes; if two opposing soldiers are on different floors of the same house, line of sight is no. Other potential uses are in the movement module.

f. MANS.HOUSE: The value of the NAME of the HOUSE in which the soldier is located. In any of the algorithms, this value is checked to eliminate those personnel not contained in the building of the same number.

g. MANS.OPENING: This value can assume a value of one to N.OPENINGS(HOUSE), assuming the combatant is in a building. By linking the combatant's opening to the direction in which he is facing, the number of line of sight checks is minimized. This is illustrated in section E of this chapter.

h. POSITION: Denotes the job position of the combatant. Although this value is not used in this model, the future usages to denote weapon type of the combatant or fire control coordinator, for example, are probable enhancements.

i. POSTURE: The value of this variable indicates the physical position of the soldier. The possible positions are standing, crouching, kneeling and prone. The amount of exposed height is adjusted by this parameter, e.g. if he is standing, his height is assumed to be two meters; if he is kneeling, his height is assumed to be one meter.

j. TITLE: The number assigned to the combatant within his own force structure.

k. X.MAN,Y.MAN,Z.MAN: The (x,y,z) coordinate of the combatant to the top of his head.

The coordinates which are input to describe the soldiers location on the urban terrain are used to determine some of his attributes, i.e. MANS.BLOCK, MANS.HOUSE, MANS.FLOOR and MANS.OPENING. Routine CALCULATE.MANS.DATA is the routine which calculates these attributes, from within the routine and by calling two other routines, FIND LOCATION (returns combatant's block and house) and LOOK.FOR.OPENING (returns combatant's opening). This routine can also be used to update these attributes of the soldier once his new coordinates have been assigned.

If a soldier is located at an opening, the routine recalculates his coordinates to allow him to view beyond the plane of the house which contains the opening. It is assumed that if a soldier is going to observe from an opening, he is going to look beyond his building in order to get a 180 degree field of view.

3. <u>Line of Sight Arrays</u>

Global arrays that are used in the line of sight routines and that are used to construct the soldier target lists are reserved and dimensioned in routine SET.UP. The line of sight arrays of each force and the individual soldiers target lists are initially set to zero. The force line of sight arrays are named RED.LOS.ARRAY and BLUE.LOS ARRAY and are dimensioned as number of blue combatants (row) by number of red combatants (column). The individual target lists are denoted RED.TGT.LIST and BLUE.TGT.LIST. As detections occur a one (1) is placed in the corresponding (i.j)th position of the force line of sight array and the pointer of the target is placed on the observer's target list.

Updates to the combatant's target lists are not made unless there has been a change in status in the corresponding force line of sight array. For example, if Red 1 detects Blue 7, but the (7,1) position in RED.LOS.ARRAY is 0, then Red 1's target list must be updated to place Blue 7 on his target list. Reversing the procedure will take a lost detection off the target list. In both cases, after the target list has been updated, the force line of sight array is updated to reflect the change in status.

C. KEY EVENTS AND ROUTINES

The driving forces in the simulation process are the events and the occurance of those events as simulated time progresses. Linked to any of these events can be other events or subroutines (routines) which control the flow of the simulation. This interaction of events with other events and/or routines can be defined as the modeller's concept of how he envisions the flow of the battle and the logical sequence in which combatant's actions occur. Section D of this chapter describes the simulation of the urban combat model. Listed below are the key events and routines in the simulation. The description of these events and routines will aid in understanding the simulation description. The source code for the events is contained in Appendix E; selected routines are described in detail in Appendix C.

a. Routine SET.UP: This routine reads in the data, assigns the attributes to the entities and initializes the line of sight arrays. Appendix B has an example and explanation of the data required to run this simulation.

b. Routine CALCULATE.MANS.DATA: This routine is called from routine SET.UP. This routine assigns the combatant's attributes based on his (x,y,z) coordinates. This routine can also be used as an update routine once the movement module is incorporated and the combatant's coordinates are changed. The following routines are called from this routine and the returned attributes are listed:

(1) Routine FIND.LOCATION: MANS.BLOCK and MANS.HOUSE

(2) Routine LOOK.FOR.OPENING: MANS.OPENING

All other attributes are assigned or calculated in routine CALCULATE.MANS.DATA or routine SET.UP.

c. Routine CHECK.LOS: This routine checks for line of sight between an observer and a target. There are a number of routines called from CHECK.LOS and which will not be described in this section. A detailed description of these routines is in Section E of this chapter. The source code and explanation of the methodology is contained in Appendix C, Section 1.

d. Routine UPDATE.TGT.LISTS: This routine updates the target lists of the combatants. The routine is called when a change in the line of sight arrays for the forces, i.e. RED.LOS.ARRAY and BLUE.LOS.ARRAY, is made. The stimulus for calling this routine is the addition of a new detection, the loss of an old detection or the killing of an opposing force combatant. The detailed explanation of the source code and the procedure for updating is in Appendix C, Section 5.

e. Event STEP.TIME: This is the driver event for this combat model. Because this is not a full combat model, line of sight is the only function really being tested. As the model matures, this event can be expanded as desired. Based on a random number draw, event LOSCHECK is scheduled for each observer-target pair. STEP.TIME recursively reschedules itself in thirty time units.

f. Event LOSCHECK: This event calls the routine to check the lines of sight between an observer-target pair as the event occurs. If line of sight does occur the routine to calculate detection times, DISMTD.CARDIO, is called and the event DETECT is scheduled.

g. Event DETECT: This event initially checks to insure that line of sight still exists between the observer and target by calling routine CHECK.LOS. If line of sight still exists, two actions occur: (1) the observer's target list is updated if necessary and (2) the event to fire at the target, KILL.MAN, is scheduled based on a random number draw for the time increment for the event to occur. Because there is currently no logic to prioritize the firer's target list, the time at which the firer engages a target is purely random.

h. Event KILL.MAN: The simulation of the firer engaging the target occurs in this event. The detailed procedure of the firing module contained in the STAR combat model, discussed in Chapter III and in [Ref. 1], is not contained in this event. In order to simplify the procedure, a simple test of kill or not kill is made. The probability that the target is killed increases as the range to the target decreases; after a Monte Carlo test, the target is either dead or alive. The actions that occur after this test is made are described in Section D of this chapter.

i. Event SIM.STOP: This calls the routines to print the target lists of all the combatants and the final status of each combatant at the end of the simulation.

D THF SIMULATION

The simulation begins once SET.UP is exited. Figure 8 shows the flow of the simulation.

The event STEP.TIME is entered immediately upon exiting SET.UP and times for lines of sight checks for all observer-target pairs are assigned. Event STEP.TIME recursively schedules itself in thirty units.

When the first line of sight event occurs, event LOSCHECK is entered. If the observer and target are both still alive, routine CHECK.LOS is entered. If line of sight does exist, routine DISMTD.CARDIO is entered and a time to detect, an incremental time from the current simulation time, is returned to the event. The detection event, DETECT, is then scheduled and control returns to the system clock.

When the detection event occurs, if the observer and target are still alive, another line of sight check is made. If the line of sight indicator returned from CHECK.LOS does not equal the current value in the line of sight array, i.e. a new detection or line of sight has been lost to an old detection, then the observers target list and the line of sight array are updated. If line of sight does exist, then

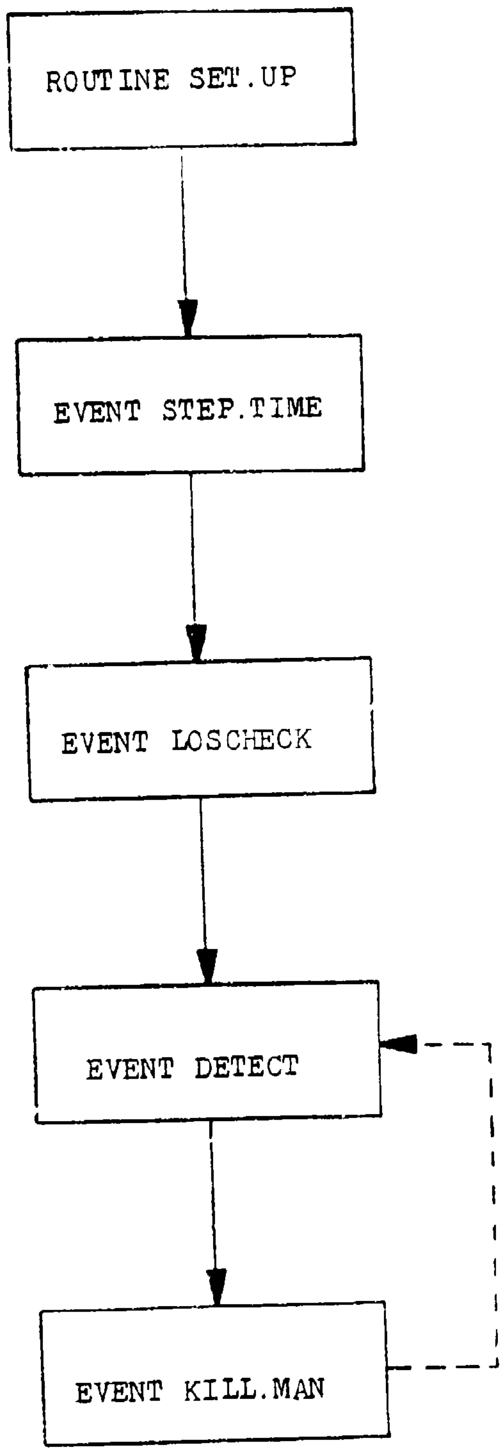


Figure 8. Simulation Flow in the Urban Combat Model

an event for the observer to shoot at the target, KILL.MAN, is scheduled and control is returned to the system clock.

When the firing event occurs, if the observer and target are still alive, the shot is fired by the observer. After conducting a Monte Carlo draw based on the range to the target, a hit or miss is assessed. If the target is hit, he is assumed to have been killed and the following events occur:

1. If the target is in a house, he is removed from that BLDG.LIST,

2. The target's activity is set to zero (soldier is dead),

3. All soldiers' target lists in the observer's army are updated to remove the dead soldier from their respective target list and the army's line of sight array is updated, and

4. The target's target list is released.

If all the Red or Blue forces have been killed the simulation is stopped, if not control returns to the system clock.

If the observer misses the target, new detection times are calculated. The observer is rescheduled for a new detection at that target, and the target is now scheduled for a detection on the observer. These new detection times are based on a random number draw, not from routine DISMTD CARDIO. If the firer has missed his target, the actions of the firer and the target are not well defined. Possible actions would be:

1. The target would go into defilade, thus blocking line of sight.

2. The target would go into defilade, but not down far enough to block line of sight.

3. The target would not be aware he was fired upon and do nothing, thus allowing the firer to reengage him immediately.

4. The target would sense the location of the firer, either visually or aurally, and then detect and engage him. These type of situations have to be investigated and modelled more thoroughly in order to create realism in this particular module.

In all these events the actual distribution of the times for the event is not taken into consideration. All events, except for time to detect, are scheduled based on a uniform random number draw; times to detect are distributed log-normal (same as in DISMTD.CARDIO) but the mean time to detection is a function only of the range to the target and the fraction visible. The enhancement of including the STAR combat model's routine, DISMTD.CARDIO, would alleviate this particular problem. The times to shoot given a detection or given a miss, still require more investigation.

The equations used in the routines to arrive at a line of sight determination have all been derived based on the rectangular structure of the buildings and the lack of a width for the representation of the soldier. The

geometrical representation of this angular battlefield did not lend itself to some of the existing techniques and routines already in current models, e.g. STAR, because of the model's use of an elliptical, bell-shaped terrain representation and the three-dimensional representation of the individual soldier. As the model matures, the representation of the soldier as a three-dimensional figure will have to be accomplished in order to produce more realistic lines of sight and casualty assessments.

E. THE LINE OF SIGHT PROCESS: AN OVERVIEW

The major portion of this urban model is comprised of the line of sight process. The major routine, CHECK.LOS, calls up to four other routines in order to return a line of sight indicator (YES/NO), a fraction visible, an observer-target range and an observer-target angle. The two subsections of this section describe the logic of the line of sight check methodology and the types of checks that need to be made in order to minimize the amount of computing required to determine if line of sight exists.

There are two assumptions made in the model for line of sight determination. These assumptions are:

(1) a combatant located inside a building not at an opening, will not observe. A combatant, inside a building and not an opening, is assumed to be doing some other activity other than observing, e.g. talking on the radio, sleeping, eating, etc., therefore his basic activity is not observing.

(2) the observer cannot see the opposite inside wall of a building through an opening. The amount of ambient light inside a room will decrease as an observer looks deeper into a room through an opening. Because of this loss of available light inside the room, it is assumed that he cannot see the opposite side of the building.

1. Routines Called in CHECK.LOS

As stated above, CHECK.LOS calls many routines in the process of determining line of sight. The routines can be categorized into three distinct functional areas:

a. To determine if line of sight can ever exist between an observer-target pair based on the direction the observer is facing and the position of the target. These routines do not take into account intervening terrain and they assume that if the target is inside of a building, but not at an opening, then it is possible for line of sight to exist.

b. To increase the efficiency of the program by returning a value that is used in another routine and

c. To calculate the values of the desired global variables.

These routines, with a brief description of the methods contained in the routine in order to perform their desired function, are listed below. A more detailed description and the source codes for these routines are contained in Appendix C.

a. CHECK.OPEN.OBS: This routine is called if the observer is in the open, i.e. on a roof or not in a building, and the target is located in a building. (If both observer and target are in the open, there exists the possibility that line of sight exists.) The routine determines the position of the observer relative to the target, e.g. north of the target or east of the target, then checks to see if the target is facing in that direction. If he is, then line of sight can exist; if he isn't, then line of sight can never exist, because the target's building is blocking the line of sight.

b. CHECK.NORTH.SOUTH: This routine is called if the observer is facing either north or south and the target is in a building. Based on the location of the observer to the target, e.g. north, southeast, west, etc., the possibility of line of sight existing can only occur if the observer-target direction pairing meets certain criteria, e.g. the observer is facing north, the target is facing south and the observer is south of the target. If the necessary condition does not exist, then the line of sight process is stopped.

c. CHECK.EAST.WEST: This routine is analogous to the above routine, except the observer is facing either east or west and the necessary conditions are also changed.

d. FIND.FACE: Assuming that the line of sight line has gone over or around a building, that building

could possibly reduce the amount of the target visible to the observer. In order to assist in this calculation, the side of the building facing the observer has to be found. This routine identifies that side of the building by utilizing the geometry of the building, i.e. rectangular in shape, and the angles induced by the observer looking at the target and at the building. The use of the trigonometric relationships of the observer, target and the building are then used to determine the side of the building facing the observer. This value is passed to routine TGT.IN.BUILDING, if the target is inside the intervening building, and is always passed to PERCENT.VISIBLE. Appendix C, Section 2 contains the more detailed explanation and source code for this routine.

e. TGT.IN.BUILDING: If the target is inside a building not at an opening, this routine is called. The existence of line of sight is determined by searching through all the openings in the target's building that are facing the observer. If there is at least one opening that offers line of sight, then PERCENT.VISIBLE is called; if line of sight does not exist, the line of sight process is stopped. The source code and detailed discussion of this routine is in Appendix C, Section 4.

f. PERCENT.VISIBLE: Once it has been determined that the observer can see the target and there is a building or opening in the vicinity of the line of sight line that could reduce the amount of the target visible to

a. CHECK.OPEN.OBS: This routine is called if the observer is in the open, i.e. on a roof or not in a building, and the target is located in a building. (If both observer and target are in the open, there exists the possibility that line of sight exists.) The routine determines the position of the observer relative to the target, e.g. north of the target or east of the target, then checks to see if the target is facing in that direction. If he is, then line of sight can exist; if he isn't, then line of sight can never exist, because the target's building is blocking the line of sight.

b. CHECK.NORTH.SOUTH: This routine is called if the observer is facing either north or south and the target is in a building. Based on the location of the observer to the target, e.g. north, southeast, west, etc., the possibility of line of sight existing can only occur if the observer-target direction pairing meets certain criteria, e.g. the observer is facing north, the target is facing south and the observer is south of the target. If the necessary condition does not exist, then the line of sight process is stopped.

c. CHECK.EAST.WEST: This routine is analogous to the above routine, except the observer is facing either east or west and the necessary conditions are also changed.

d. FIND.FACE: Assuming that the line of sight line has gone over or around a building, that building

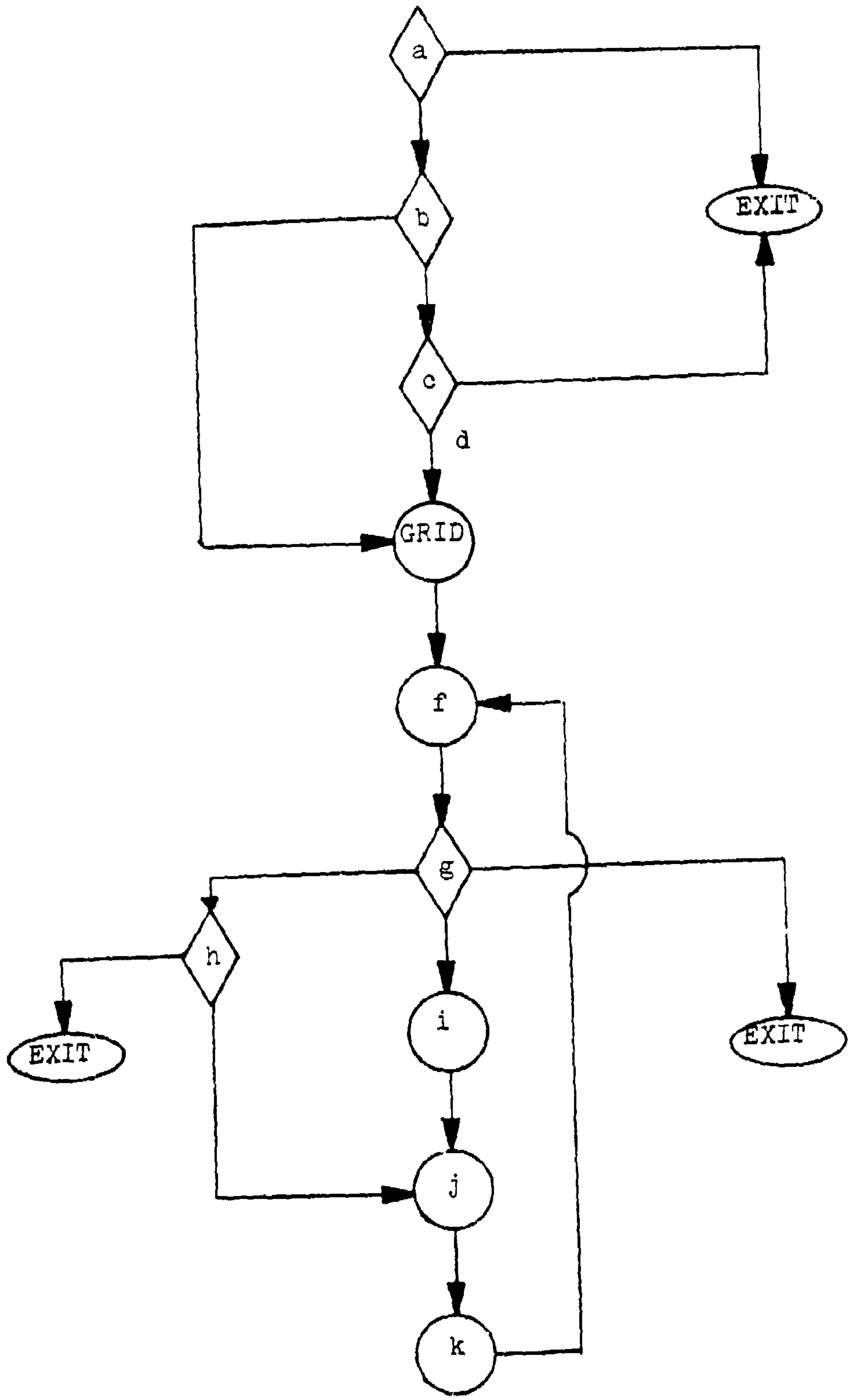


Figure 9: Line of Sight Flow in the Urban Combat Model

c. Otherwise the flow is directed to the appropriate routine based on the observer's status, i.e. in the open, facing north or south or facing east or west. These subroutines are CHECK.OPEN.OBS, CHECK.NORTH.SOUTH and CHECK.EAST.WEST. All three subroutines return a line of sight indicator. If it is NO, then the positioning of the target relative to the observer prevents any occurrance of a line of sight and the routine, CHECK.LOS, is exited.

d. Otherwise the flow goes to control point GRID.

e. At GRID, the number of blocks crossed by the line of sight, as well as the block designations, are determined. The technique to determine these blocks is taken from [Ref. 5]. The purpose of the procedure is to reduce the number of buildings that have to be checked for line of sight interference, thus reducing CPU and run time on the simulation.

f. The block numbers are passed to two loops within which a search along the line of sight is conducted to determine if a building impedes the observer's view of the target. Only those buildings within the designated block are checked.

g. If an intervening building blocks the line of sight and the target is not inside that house, then line of sight does not exist and the routine is exited.

h. If the intervening building is the target's, then the routine FACE.OBS is entered. The output from FACE OBS is then input to routine TGT.IN.BUILDING, which determines line of sight by looking through the openings of the building facing the observer. If line of sight is blocked, the routine is exited.

i. If the current building does not impede the line of sight and the line of sight line passes over the building, then FACE.OBS is entered. In order to determine whether or not the line of sight breaks the side planes of the building, i.e. passes over the building, a check is made before FACE.OBS is entered. The intent of this check is to minimize the number of buildings that have to be checked for possible degradation of the fraction of the target visible to the observer.

j. In both cases, h and i above, the routine PERCENT.VISIBLE is entered and the fraction of the target visible to the observer is returned.

k. The variable PER.CENT.VISIBLE is then updated by taking the minimum of the previous value of PER.CENT.VISIBLE and the value returned from PERCENT.VISIBLE. The flow then returns to check the next building until the loops are exhausted.

l. If line of sight exists, the range to the target (RANGE.TO.TARGET) is calculated and the routine is exited. The two global variables PER.CENT.VISIBLE and RANGE.TO.TARGET are both used in the routine DISMTD.CARDIO.

The line of sight procedure is comprised of many checks, and if not done efficiently, unnecessary computer time may be used. The procedures done in this routine that have reduced the computer run time are summarized below:

a. The initial screening process in this routine, steps a to c above, eliminates those observer-target pairs in which line of sight is obviously YES or NO.

b. The isolation of the number grid squares crossed by the line of sight line, reduces the number of buildings that have to be checked.

c. The calculation of the fraction visible for only those buildings that are along the line of sight reduces the number of checks.

d. The locating of the side of the building facing the observer, given the target is inside the building, and then checking only those openings on that side of the building for line of sight also reduces the number of checks.

V. FUTURE ENHANCEMENTS

This thesis presents a foundation for an urban combat model on which a complete model can be built. Although it is a stand alone model, the capability to directly use some of the routines and events from the STAR combat model, as discussed in Chapter III, will save modelling time. Other areas of combat, which are strictly urban combat related or not included in the dismounted STAR model, are still needed if one wants a higher resolution model. This chapter will delineate some of the future enhancements which are of near term interest in order to quickly get a working urban model.

A. MOVEMENT LOGIC

Movement in the urban terrain is characterized by a combatant darting from one covered and/or concealed area to another or by exposing himself for only very short periods of time for observing or firing. Other types of movement, discussed in the URBWAR model include movement to gain a more advantageous position for observing or firing.

The movement model can incorporate any of two possible models, preplanned movement as in STAR or dynamic route selection as reported in DYNTACS or ASARS. Preplanned routes can be employed for moving a unit into an urban area. However, once contact is made, the movements of each combatant can no longer be preplanned and based on some

criteria, they will assault or defend in the most effective manner.

1. STAR Movement (Preplanned)

As discussed in Section IIIC, movement in the dismounted infantry version of STAR is nodular in structure from start to end points. Movement in this manner is characterized by a set formation, input value, from turn point to turn point, until the simulation has stopped. It is not realistic to attempt to model this type of movement in urban terrain (the realistic value of this type of dismounted movement in any type of terrain is questionable) except to initially move an element into the urban terrain and to designate the subobjectives and final objective for the assaulting element until contact is made. At this time the dynamics of characteristic urban combat must dictate the movement of the combatants until the simulation is over.

2. Dynamic Movement

No explicit recommendations for an approach to modelling dynamic movement will be discussed in this section. It is unclear what prerequisite conditions must exist before a combatant will perform a certain action. The analysis of all the stimuli that would invoke movement and the proper movement to take are the critical steps in this module.

The stimuli that can invoke movement can be of three types:

a. Orders from a superior to move to point B. This unfortunately reveals another source of modelling dilemmas, i.e. the leaders decision-making process.

b. Visual detection of danger. This can take on many forms such as the muzzle flash, the striking of a round in the vicinity of an enemy soldier but out of the effective range of his weapon or a glimpse of a darting soldier across an opening. An interpretation as to the criteria for a danger state is required to filter those states the soldier may be in that are not considered dangerous.

c. Aural cues. The presence of gunfire will always force a combatant to search in the direction of the stimulus. In the absence of orders, if the gunfire is close to the soldier it is natural to move to the sound to seek its source.

When a person moves through an urban area, certain aspects of the terrain structure must be considered. When moving through an area with knowledge of enemy in the area, this knowledge coming from intelligence reports or explicit cues to their presence, avoidance of open areas and quick darting movements from covered area to covered area is the preferred method. The process of modelling this type of movement has to take into consideration:

a. The location of the nearest opening relative to the combatant's direction of movement.

b. The location of those areas known to be controlled by the enemy either by fire or observation.

c. The distance separating the combatants of the same force. (An input parameter that must be considered to simulate some level of control of the force).

d. The locations of corners of buildings to force the combatant to halt, observe around the corner and proceed if no danger exists. (An implicit modelling consideration is how to model only a portion of the soldiers head as he looks around or over his covered area.)

e. If modelled, the location of obstacles, e.g. cars, rubble, etc., for the combatant to hide behind, given he must move in that direction and the obstacle offers sufficient cover from fire and observation.

f. Combat in urban terrain does not dictate that the shortest distance between two points is a straight line. Evasive, darting, zig-zagging movement is often characteristic.

Movement in the interior of a building for the both the attacker and the defender is also dynamically dictated.

a. Location of the nearest opening given the combatant is cued to the presence of an enemy in a certain direction.

b. Location of the stairwell.

c. Controlling the speed of a combatant going up or down a stairwell depending on the known enemy situation.

d. Moving from an opening given a round has struck in his vicinity, and then relocating at another opening that offers him the same field of view in order to search for and engage the enemy.

B. TIMES TO DETECT

1. General

The current simulation does not contain a valid routine to return detection times. A possible choice is the routine DISMTD.CARDIO, discussed in Chapter III. The URBWAR combat model eludes to another procedure for detection times for a non-firing target. Reference is made to a glimpse time and although the narrative found in [Ref. 5] does not explicitly state a modelling approach other than the vision lobe theory, the glimpse model for detection is probably used in the model. This next section will discuss the glimpse modelling approach for detection.

2. The Glimpse Model

A glimpse is defined as one visual fixation when given intermittent chances to view a target scene. This visual fixation is approximately equal to 1/4 of a second. The value of this type of modelling procedure in an urban model, is the numerous potential occurrances of a target dashing across an opening or across a street in which the exposure time of the target is considerably less than a target in an open field.

The modelling procedure defines the following variables:

a. n = number of glimpses until detection

b. g(i) = P(target is detected on the ith glimpse)

Therefore,

$$P(\text{ failed to detect on the 1st n-1 glimpses}) = \prod_{i=1}^{n-1} (1 - g(i))$$

$$P(\text{ 1st detection on nth glimpse}) = g(n) \prod_{i=1}^{n-1} (1-g(i)).$$

Assuming that g(i) is equal for all i = 1,2,...,n

P(1st detection on nth glimpse) = g * (1-g)**(n-1)

which is the density function for a geometrically distributed random variable. The value of E(n), the expected number of flimpses until first detection, is equal to 1/g and the maximum likelihood estimator of g is equal to 1/(mean number of glimpses until first detection).

When used in a discrete event simulation, there are two possible procedures for using the glimpse model:

1.) Let a glimpse be an event

a. Draw a uniform (0,1) random variable = u1

b. If u1 < g, a detection has occurred

2.) Let the detection be an event.

a. Draw a uniform (o,1) random variable = u2

b. Using the probability integral transform for the cumulative distribution function of a geometric random variable, solve for n, the number of glimpses until the detection occurred; with u2 = P(N > n), N being distributed geometrically with parameter g.

c. Therefore,

Time to Detect = n * (time per glimpse)

or

Time to Detect = n / (glimpse rate)

The value of the parameter g is a function of a number of factors, such as movement of the observer and/or target, complexity of the scenary, target background, ambient light level and the cueing stimulus. If one were to use this type of model and there were m different factors for every value of g, an m-dimensional ragged array would have to be constructed, stored and accessed for a table look-up technique for finding g. A ragged array is used in the SIMSCRIPT II.5 simulation language. Unlike an m by n array, where the number of elements in row i is equal to n for all i=1,2,...,m; in a ragged array, the number of elements per row are not necessarily equal.

The feasibility of this type of procedure, although a time consuming data input procedure, is well within the capabilities of modern day computers, and this modelling approach is offered as an adjunct to the detection model for consideration.

C. PRIORITY OF ENGAGEMENT LOGIC

During a combat simulation, a combatant can have multiple detections. In the STAR combat model, an event TARGET.SELECT is scheduled in order to select the "best" target for that combatant to engage before a firing event can be scheduled.

If the combatant is to select a highest priority target, a list of criteria must be followed in order to rank order all those targets he has detected before a firing event is scheduled. A follow-on action, not modelled in this urban combat model, is the selection of the correct weapon with which to engage the target, once the target is selected.

Examples of rank ordering criteria include:

1. By weapon type of the enemy if known.
2. By closest enemy.
3. By leadership position of enemy if known.
4. By enemy last engaging combatant.
5. By order of higher leader.
6. By targets not being engaged by friendly units.

The STAR combat model allows the user to input his priority of target selection. To further enhance this urban model, once the criteria is established by the modeller, the same option should be given the user for target selection.

A follow-on action, not modelled in this urban combat model, is the selection of the correct weapon with which to engage the target, once the target is selected. In order to model the method of engagement, criteria must be established to insure the combatant selects the correct weapon for the correct target in the given situation. The delineation of the necessary conditions can be a time-consuming task, but it is necessary for the realism of the situation.

D. RESOLUTION - HOW MUCH IS ENOUGH

The above future enhancements are the stepping stones to the completion of this model. As a long term project to further enrich the model to a higher resolution, the obvious macro and the subtle micro details of characteristic urban terrain will have to be modelled. Cellars, subways and sewers are examples of the macro enrichments; walls, ventilator shafts and interior doors are examples of micro enrichments. These enhancements will lead to a more realistic simulation, but one has to consider the price for the enrichment.

Characteristic of large simulations is the increase in central processing units (CPU), run times, core requirements for storage, disks and tapes. When one approaches the enriching of a model, these factors must be weighted against the increase in realism of the simulation. Some approaches to determining the worth of a higher resolution model can include:

1. Comparing the payoffs for higher resolution vice the price for the enrichment,

2. Comparing the analytical results of a simulation with the higher resolution vice the same simulation without the increase in resolution, or

3. Asking the proponent agency requesting the model how much resolution is required for their study. The question they may want answered may not include some of the characteristics of urban terrain.

APPENDIX A. DEFINITION OF SETS, ENTITIES, VARIABLES, ROUTINES AND EVENTS USED IN THIS MODEL

SETS

TOWN	Contains the pointers of all houses in the simulation.
BLDG.LIST	Contains the pointers of all soldiers in the house owning this set.

ATTRIBUTES OF TEMPORARY ENTITY HOUSE

BLOCK	Integer value which designates the block in which the house is located.
DIM.FLOOR	Distance between floors in the house. (assumed constant in this model)
DIM.X	Dimension in the x-direction
DIM.Y	Dimension in the y-direction
DIM.Z	Dimension in the z-direction
KM	Constant assigned to the house based on the structural material. The constant is used in determining if a round penetrated a building.
NAME	Integer value for the house number. Houses are numbered sequentially from 1 to NN.HOUSE.
N OPENINGS	Integer value for the number of openings in the house.
X.1,X.2,X.3,X.4	X-coordinate values of the four corners of the house starting in the lower-left corner (southwest) and moving counter-clockwise.

Y.1,Y.2,Y.3,Y.4	Analogous to the x-coordinates.
THICK	Thickness of the structure in inches. Assumed constant throughout the entire structure.
TYPE	Integer value for the type of building material. The building can have four basic structural types: 1. Wood 2. Concrete 3. Reinforced concrete 4. Metal

ATTRIBUTES OF TEMPORARY ENTITY MAN

ACTIVITY	Integer value for the type of activity in which the soldier is engaged, i.e.; 0. Dead 1. Observing 2. Firing 3. Being engaged 4. Moving 5. Being engaged and moving
ARMY	Integer value for the soldier's army. 1. Red 2. Blue
MANS.BLOCK	Integer value for the block in which the soldier is located. If the soldier is not in the grid system, this value is set to zero.
MANS.DIR	Integer value for the direction in which the soldier looking. If he is in an opening, it is set to the appropriate value, i.e.: 1. South, 2. East, 3. North or

4. West.

If he is on a roof or in the open it is set to 999; otherwise if he is in a house but not at an opening it is set to zero.

MANS.FLOOR	Integer value for the floor of the house on which the soldier is located. If he is in the open, this value is set to zero.
MANS.HOUSE	Integer value for the number of the house in which the soldier is located, if he is not in/on a house, it is set to zero.
MANS.OPENING	Integer value for the number opening that the soldier is located. If he is not at an opening, this value is set to zero.
POSITION	Integer value for the soldier's position in his respective organization, i.e.: 1. Platoon leader 2. Squad leader 3. Team leader 4. Rifleman

POSTURE	Integer value describing the type of body position the soldier has assumed, i.e.: 1. Standing (2 meters) 2. Crouching (1.5 meters) 3. Kneeling (1.0 meters) 4. Prone (0.5 meters) The parenthetical entries relate the amount of the soldier exposed in the z-direction, given he has assumed that posture.
TITLE	Integer value assigned to the soldier. Soldiers are numbered sequentially from 1 to NN.MEN.
X.MAN,Y.MAN,Z.MAN	Coordinates to the top of the soldier.

GLOBAL VARIABLES

BLK.GRID	Integer array of dimensions NBLKX by NBLKY. The (ij)th element is the number of the block. Blocks are numbered from 1 to NBLKX*NBLKY starting at the origin, along the Y-dimension for all X and then for all Y.
BLKSIZE	Dimension of the block (grid). Blocks are modelled as squares.
BLUE.LOS.ARRAY	Integer array of dimensions NN.BLUE by NN.RED. The (ij)th element contains the binary value for line of sight, i.e. 0=NO, 1=YES.
BLUE.TGT.LIST	Integer array of variable dimension. The ith element is the pointer of the Red soldier. If the list is empty, i.e. no detections, then the first element is set to zero. If the Blue soldier who owns the list is dead, the array is released.
DEAD.BLUE	Integer variable that counts the number of Blue soldiers dead.
DEAD.RED	Analogous to DEAD.BLUE.

NBLKX	Integer variable for the number of blocks in the X-direction in the terrain.
NBLKY	Analogous to NBLKX.
NN.BLUE	Integer variable for the number of Blue soldiers in the simulation.
NN.HOUSE	Integer variable for the number of houses (building) in the simulation.
NN.MEN	Integer variable for the total number of soldiers in the simulation.
NN.RED	Analogous to NN.BLUE.
OBS.TGT.ANGLE	Measure of the angle from the observer to the target in the XY-plane.
OPEN.DIMEN	Real array of dimensions NN.HOUSE by 3 * N.OPENINGS. The elements of the array are input in groups of three. The first element is the dimension of the opening in the X-direction, the second in the Y and the third in the Z-direction. All dimensions are offsets from the lower left coordinate of the jth opening.

Variable N.OPENINGS is an attribute of the HOUSE and is the number of openings.

OPEN.FACE	Integer array of dimensions NN.HOUSE by N.OPENINGS. The (ij)th element is the integer which indicates which direction the opening is facing, i.e. 1. South 2. East 3. North 4. West
OPENING.ARRAY	Real array of dimensions NN.HOUSE by 3 * N.OPENINGS. The inputs are similar to OPEN.DIMEN, except the entries are the coordinates of the lower left corner of the opening.
PER.CENT.VISIBLE	Real variable indicating the fraction of the target visible to the observer. This fraction is calculated assuming the soldier is exactly two meters tall.
PERS.BLUE.LIST	Integer array of dimensions NN.BLUE by 2. The (i,1) element is the pointer to the ith Blue soldier's

	target list; the (i,2) element is the pointer to the ith Blue soldier.
PERS.RED.LIST	Analogous to PERS.BLUE.LIST.
PT.HOUSE	Integer array of dimension NN.HOUSE. The ith element is the pointer to the ith HOUSE.
PT.MAN	Integer array of dimension NN.MEN. The ith element is the pointer to the ith MAN.
RANGE.TO.TARGET	Real variable indicating the distance from the observer to the target.
RED.LOS.ARRAY	Analogous to BLUE.LOS.ARRAY.
RED.TGT.LIST	Analogous to RED.TGT.LIST.

ROUTINES

CALCULATE.MANS.DATA	Calculates/updates the attributes of the soldier.
CHECK.EAST.WEST	Determines if line of sight exists, given the observer is facing either east or west.
CHECK.LOS	Determines if line of sight exists.
CHECK.NORTH.SOUTH	Determines if line of sight exists, given the observer is facing north or south.
CHECK.OPEN.OBS	Determines if line of sight exists, given the observer is in the open or on a roof.
DISMTD.CARDIO	Returns a time to detection.
FIND.FACE	Returns the side of a building facing the observer.
FIND.LOCATION	Returns the number of the block and the number of the building of the soldier.
LOOK.FOR.OPENING	Returns the number of the opening at which the soldier is located.

PERCENT.VISIBLE	Returns the fraction visible of the target to the observer, i.e. global variable PER.CENT.VISIBLE. The global variable OBS.TGT.ANGLE is also calculated in this routine.
SET.UP	Initializes all entities and global variables for the simulation. The routine is released prior to the start of the simulation.
TGT.IN.BUILDING	Determine if line of sight exists, given the target is located inside a building.
UPDATE.TGT.LISTS	Updates the target list of the observer, given there has been a change in status of his line of sight to the target.

EVENTS

DETECT	1. Calls routine CHECK.LOS. 2. Updates the target list of the observer. 3. Schedules event KILL.MAN if line of sight exists.
KILL.MAN	1. Determines if the target was killed. 2. If the observer missed, event DETECT is scheduled for the observer to the target and for the target to the observer. 3. If the target was killed, all target lists of the observer's army are updated, given the target was on their target list. 4. If all the soldiers on the target's army are dead, the simulation is stopped.
LOSCHECK	1. Calls routine CHECK.LOS. 2. If line of sight exists, routine DISTMD.CARDIO is called and event DETECT is scheduled. 3. If line of sight does not exist the event is exited.
SIM.STOP	1. Prints the final target lists. 2. Prints the attributes of the soldiers. 3. Stops the simulation.

STEP.TIME 1. Schedules LOSCHECK for all pairs of Red and Blue forces.

2. Schedules itself recursively.

APPENDIX B. EXPLANATION OF DATA INPUT

The data is free-format input. The table below indicates at which place in the data file each input is placed. Explanations off to the side or on top of each entry are provided for clarity.

DATA FOR ENTITY HOUSE

```
 2     2    200.                          -NNBLKX, NNBLKY, BLK.SIZE
 1                                        -NN.HOUSES
BLOCK       X.1     DIM.X     DIM.Z        NN.OPENINGS
    HOUSE       Y.1      DIM.Y     DIM.FLOOR          TYPE
  1   1      50.  50.  50.  50.  20.    5.       12      2
4 4 4 4 3 3 3 3 2 2 2 2                  -ELEMENTS OF OPEN.FACE
 00.   0.   25.   2.     1.   2.    -ELEMENTS OF OPENING.ARRAY
 00.   0.   25.   2.     6.   2.    1.  OFFSET IN X-DIRECTION
 00.   0.   25.   2.    11.   2.    2.  DIMENSION IN X-DIR
 00.   0.   25.   2.    16.   2.    3.  OFFSET IN Y-DIRECTION
 25.   2.   50.   0.     1.   2.    4.  DIMENSION IN Y-DIR
 25.   2.   50.   0.     6.   2.    5.  OFFSET IN Z-DIRECTION
 25.   2.   50.   0.    11.   2.    6.  DIMENSION IN Z-DIR
 25.   2.   50.   0.    16.   2.
 50.   0.   25.   2.     1.   2.
 50.   0.   25.   2.     6.   2.
 50.   0.   25.   2.    11.   2.
 50.   0.   25.   2.    16.   2.
 10.                                   -THICKNESS OF STRUCTURE
```

DATA FOR ENTITIES MAN

5 3 2 -NN.MEN, NN.RED, NN.BLUE

TITLE	ARMY	POSITION	X.MAN	Y.MAN	Z.MAN	POSTURE	ACTIVITY
1	1	3	250.	76.	17.0	1	1
2	1	4	255.	76.	21.0	3	3
3	1	4	260.	76.	21.5	1	2
1	2	3	100.	326.	22.0	1	1
2	2	4	101.	250.	.5	4	4

APPENDIX C. SELECTED ROUTINES IN THE URBAN MODEL

1. Routine CHECK.LOS

Routine CHECK.LOS requires two arguments to be passed into the routine and one value will be returned. The passed arguments are the pointers of the observer and the target, the returned value is the line of sight indicator (0 = no, 1 = yes). A copy of the source program is attached and references to this and the corresponding line numbers will be made in this discussion. (See page 131)

There are a series of checks throughout this routine that were designed to screen out all those observer-target lines of sight that do not require the more extensive screening process which starts at control point GRID (33). These screens include:

1. If the observer and target are on the same floor in the same building, there exists line of sight. (4-6)

2. If either the observer or target is not in the grid coordinate system there is no line of sight. (7)

3. If the observer is inside a building but not at an opening then the soldier does not observe. (8) It is assumed in this model that a soldier inside a building but not at an opening, is performing some other function other than observing and he will never perform the task of observing while in this position.

4. If the observer and target are in the same building, but on different floors, there is no line of sight. (9-10) The task of modelling stairways was not accomplished. That degree of resolution is best reserved as a future enhancement.

At this point in the program observers are either in the open, on the roof of a house or in an opening and targets are anywhere within the grid coordinate system.

The local parameters X.DIFF, Y.DIFF and Z.DIFF are the differences in the coordinates of the target to the observer. In order to eliminate the possibility of dividing by zero later in the program, if the value of any of the values is exactly zero, it is set to an arbitrarily small number. (12-17)

A second set of screens were included in this routine to further reduce the input down to GRID. In all cases if these screens do not result in a line of sight indicator of no, the flow of the program goes to GRID. This second set of screens include:

1. If the target is in the open or on a roof, proceed to GRID. (22) It is assumed that any target not under any type of cover or concealment is a likely candidate for a line of sight.

2. If the observer is in the open or on a roof, enter into routine CHECK.OPEN.OBS. (25)

3. If the observer is facing either north or south, enter into routine CHECK.NORTH.SOUTH. (28-29)

4. If the observer is facing either east or west, enter into routine CHECK.EAST.WEST. (30)

The details of how each of the above three routines determine line of sight will not be dealt with in this document. The source codes are included. (pages 135-137)

To enter into the last two routines the observer has to be in a building observing out of an opening. In general terms, the basic principle used in determining line of sight is, if the location of the observer relative to the target is such that his (the observer's) building is blocking his view to the target, then line of sight does not exist.

If the observer is in the open, the target's position relative to the observer's is used in the determination of line of sight. It is assumed that if a target is looking a certain direction from an opening and the observer is looking in the same direction as the target, that even though there is a possibility that line of sight can occur through an opening directly opposite the target, it is prohibited by the lack of light present inside of the room to illuminate the target sufficiently for line of sight. Hence, if the target is in an opening, line of sight can only occur if the observer is within the 180 degree field of view of the target.

The next step in the procedure is to search along the observer-target line to see if any of the buildings impede

the line of sight. In order to minimize the number of buildings that have to be checked, it is necessary to determine the grid squares that are crossed by the observer-target line of sight. The logic and procedures contained in lines 34 to 63 were extracted from Professor James K. Hartman [Ref. 5]. The program was rewritten in part to accomodate this model.

The first entry into the grid square list is the block indicator, IX and IY, which contains the target and is entered as IGX(1) and IGY(1). The variables ISGX and ISGY are set to (+/-) one depending on whether the grid square indices, IX and IY, increase or decrease as the search is conducted from the target to the observer. The variables XSTEP and YSTEP are set to be the fraction of the distances X.DIFF and Y.DIFF moved on from the target to the observer. XINC and YINC increment XSTEP and YSTEP as the line of sight crosses one grid square in the X and Y directions.

In order to continue the search into the next grid square the variables XSTEP and YSTEP are set to the intersection of the next X-grid intersection and Y-grid intersection. The smaller of the two values cues the procedure as to which intersection is encountered first and hence, which variable IX or IY will be incremented by ISGX or ISGY respectively. After each increment, the new grid square is added to the IGX and IGY arrays. When both XSTEP and YSTEP are greater than 1.0, the search is stopped (it has proceeded beyond the target), and the grid squares crossed by the line of sight are all recorded in the IGX and IGY arrays.

The two loops, lines 67 to 110, contain the major checks for the line of sight determination. The outer loop increments through the grid squares found in the above procedure and the inner loop increments through all the houses contained in the current grid square. The variable PER.CENT.VISIBLE is initially set to one and is updated as the check for the fraction visible of the target to the observer is made.

The observer-target line has been parametrically represented in three-space as:

$$X = X(tgt) + S * (X(tgt) - X(obs))$$
$$Y = Y(tgt) + S * (Y(tgt) - Y(obs))$$
$$Z = Z(tgt) + S * (Z(tgt) - Z(obs))$$

expressed in terms of a single parameter, S. The values (X,Y,Z) represent either the maximum or minimum values of the buildings coordinates depending on which portion of the checking process the program is executing.

Apricri, the variable SMIN is set to 0.0 and the variable SMAX to 1.0. These two values represent the minimum and maximum values of S; the point S = 0.0 is at the target and S = 1.0 the observer. In order to solve for the variable S, the equations become:

$$S = (X - X(tgt)) / (X(tgt) - X(obs))$$
$$S = (Y - Y(tgt)) / (Y(tgt) - Y(obs))$$
$$S = (Z - Z(tgt)) / (Z(tgt) - Z(obs))$$

The procedure for solving for and updating the values of SMIN and SMAX are identical for all three planes (71-86). Only that section dealing in the X-plane will be discussed.

The variables SLO and SMAX are defined as:

SLO = (SMIN(house) - X(tgt)) / X.DIFF

SHI = (SMAX(house) - X(tgt)) / X.DIFF

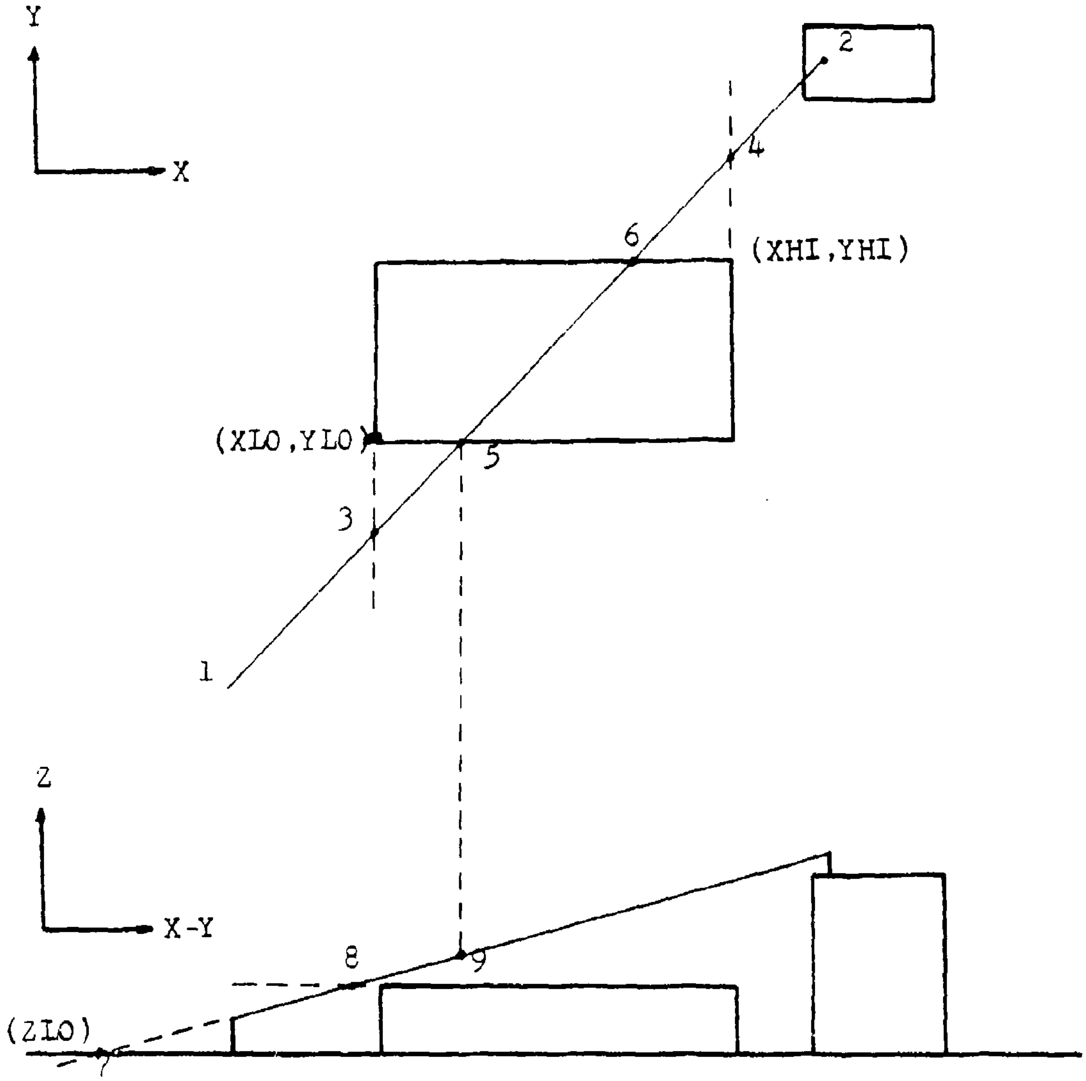
These values represent the changing of the minimum and maximum values of S as one proceeds along the observer-target line until it intersects the X-plane of the building. To update these values, the variables S1 and S2 are defined to be the minimum and maximum of SLO and SHI respectively. In order to move the points SMIN and SMAX along the line, the maximum of SMIN and S1 are assigned to SMIN, and the minimum of SMAX and S2 are assigned to SMAX.

This basic procedure is followed through all three planes of the building. If the final value of SMIN is less than or equal to SMAX, the building has intersected the observer-target line; if it isn't, the flow returns to check the next building in that grid square until all the buildings in all the designated grid squares have been checked.

Figures C-1 is an example of the entire process. The following is an explanation of the numbering sequence depicted in the figure:

1. The location of the target and the point at which SMIN = 0.0.

2. The location of the observer and the point at which SMAX = 1.0.



(1) Numbers 1,3,5, and 9 indicate the sequential movement of SMIN along the line of sight.

(2) Numbers 2,4,6 and 8 indicate the sequential movement of SMAX along the line of sight.

(3) Calculations are made in the X, Y and Z planes respectively in that order.

SMIN > SMAX

therefore

Line of Sight = YES

Figure C-1. Determination of Line of Sight

3. Intersection with XLO; defines S1 and also SMIN.

4. Intersection with XHI: defines S2 and also SMAX.

5. Intersection with YLO; defines S1 and also SMIN.

6. Intersection with YHI; defines S2 and also SMAX.

7. Intersection with ZLO; defines S1 but not SMIN.

The value of SMIN from the y-axis is greater than S1, therefore, SMIN is unchanged.

8. Intersection with ZHI; defines S2 and also SMAX.

9. The value of SMIN is greater than SMAX, therefore line of sight exists.

The program follows two different paths depending on whether or not a particular building impedes the line of sight.

THE BUILDING DID NOT IMPEDE THE LINE OF SIGHT

The variable IFLAG is initially set to zero, line 70. The function of this variable is to count the number of times the relationship of SMIN and SMAX meets a certain criteria. In the X and Y-planes, if SMIN is less than SMAX then IFLAG is incremented; in the Z-plane if SMIN is greater than SMAX then IFLAG is incremented. (lines 76, 81, 86 respectively) After checking all three planes, if the value of IFLAG is equal to three (88), then the observer-target line of sight has crossed the X and Y-planes of the building but not the Z-plane, therefore that building has to be checked to see what fraction of the target has been blocked by that building. If IFLAG is not equal to three, the next building in that grid square is checked.

In order to calculate the fraction visible of a target, a preliminary step must be taken before entrance into routine PERCENT.VISIBLE. The routine FIND.FACE (90) will find the side of the building that is facing the observer. The output of FIND.FACE is input to PERCENT.VISIBLE (93) and the routine returns the fraction visible of the target, defined as PER.CENT.VISIBLE. The minimum of this value and the previous value of the variable is then set to PER.CENT.VISIBLE (95) and the next building is checked. Once all buildings in the designated grid squares have been checked and there is line of sight to the target, the global variable RANGE.TO.TARGET is calculated (112) and the routine is exited with an indicator of YES for line of sight.

THE BUILDING IMPEDES THE LINE OF SIGHT

If the building impedes the observer's line of sight, a check is made to determine if that building is the same building as the target's (98). If is not, then line of sight has been broken and the routine is exited with an indicator of line of sight NO. Otherwise, the target has now been located in that building and the routine FIND.FACE is entered (100). The output from this routine is entered into routine TGT.IN.BUILDING (102). This routine searches through all the openings of the building on the side facing the observer. If the target can be seen through an opening, an indicator of YES is returned and the program

proceeds to the next routine PER.CENT.VISIBLE (105). From this point, the flow of the program is the same as described above. If there is no opening through which the observer can see the target, an indicator of NO is returned and the routine is exited.

2. Routine FIND.FACE

The function of this routine is to find the side of the building that is facing the observer. The input parameters for this routine are the pointers of the observer, target, the differences in the coordinates of the observer and target (observer - target), and the pointer of the building being checked. The output variable is the side of the building facing the observer. This value is used as an input parameter to the routines PERCENT.VISIBLE and TGT.IN.BUILDING. The source code for this routine is attached. References to line numbers throughout the source code will be made in the text of this discussion. (See page 138)

The observer-target angle and the angle the observer makes with a specific corner of the building are the two major parameters in this routine. The two angles are measured in the XY-plane and the arctangent function returns an angle between 0.0 and pi radians or an angle between 0.0 and minus pi radians. Figures C-2 and C-3 show the position of the observer, target and the buildings being checked. References to the observer, target, buildings and other variables in this text, can be located on these figures.

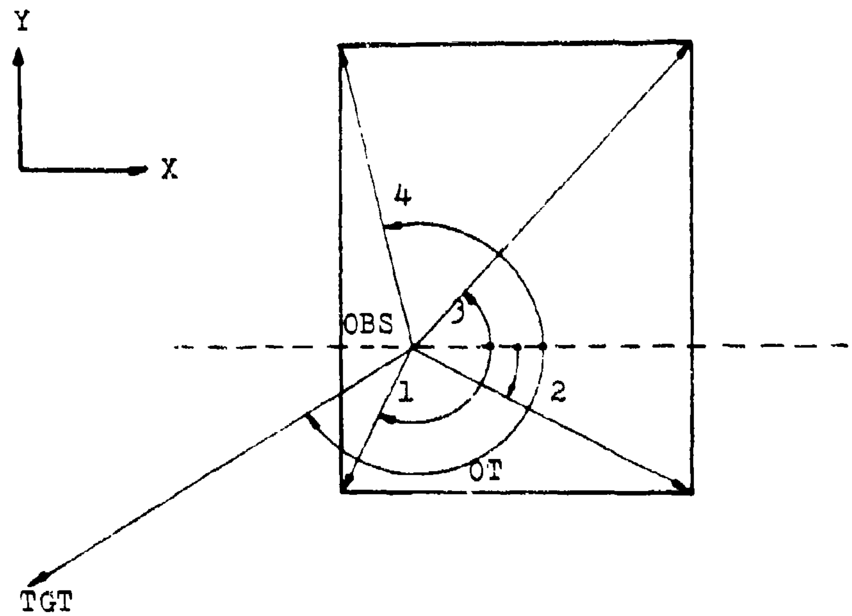


DIAGRAM ANGLE	PROGRAM ANGLE
1	A1
2	A2
3	A3
4	A4
OT	OBS.TGT.ANGLE

A4 < OBS.TGT.ANGLE < A1

THEREFORE

FACE.OBS = 4

Figure C-2. Side of the Building with Observer on the Roof

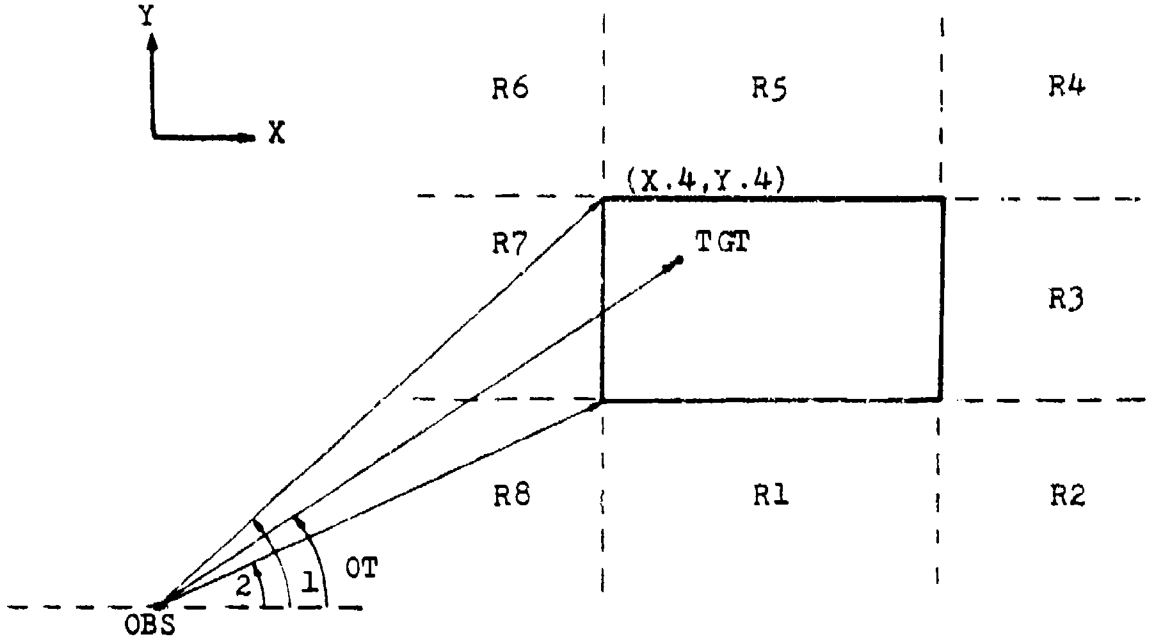


DIAGRAM REGION	PROGRAM
R1,R5	CHECK.Y
R2	CHECK.TWO.ONE
R3,R7	CHECK.X
R4	CHECK.THREE.TWO
R6	CHECK.FOUR.THREE
*R8	CHECK.ONE.FOUR

*Observer is in region 8, therefore he is facing either the south or west side of the building.

ANGLE 1 = ARCTAN((Y.1-YOBS)/(X.1-XOBS))
ANGLE 2 = ARCTAN((Y.4-YOBS)/(X.4-XOBS))

ANGLE 1 < OBS.TGT.ANGLE < ANGLE 2

THEREFORE

FACE.OBS = 4

Figure C-3. Side of the Building with Target on the Roof

The procedure followed in this routine makes use of the geometry of the building being represented as a rectangular figure as projected in the XY-plane. The arctangent function is used to determine all the angles required.

THE BUILDING BEING CHECKED IS THE OBSERVERS

Figure C-2 illustrates this case. The angles A1, A2, A3 and A4 are calculated (5-10). These angles are used as a means of locating the observer on the building relative to the four corners of the roof. By locating the observer-target angle between any adjacent pairings of the above calculated angles (11-17), the side of the building facing the observer is found and the routine is exited.

THE BUILDING BEING CHECKED IS NOT THE OBSERVERS

In order to locate the specific side of the building in question, the procedure is broken into four steps (Figure C-3):

1. Place the observer into one of eight possible locations in the XY-plane (19-30). These locations are specified by the coordinates of the building. This initial check decreases the number of available sides of the building that are possible for the observer to be facing.
2. Proceed to the part of the program which isolates at most two sides of the building that the observer can possibly be facing.

3. Compare the observer-target angle to the angle the observer makes with the specific corner of the building which describes that side of the building (32-57).

4. The result of the check is the side of the building facing the observer and the routine is exited.

3. Routine PERCENT.VISIBLE

This routine is used to calculate the fraction visible of the target to the observer, given that a building or an opening is intersecting the line of sight in both the X and the Y-planes. The input parameters to this routine are the pointers of the observer, target, building, the differences in the coordinates between the observer and target, the elevation of the object being checked, e.g. the elevation of the building or the lower dimension of the opening, and the side of the building facing the observer. The resulting output value is the fraction visible of the target. This fraction is based on all targets being exactly two meters high. This program does not model the individual soldier in three dimensions and the observer must see the top of the target before line of sight can occur, therefore this fraction visible is from the top of the target down to the lowest point of the target visible to the observer.

The source code for this routine is attached. References to specific lines of the code will be made throughout this text. Figures C-4 and C-5a depict the location of the

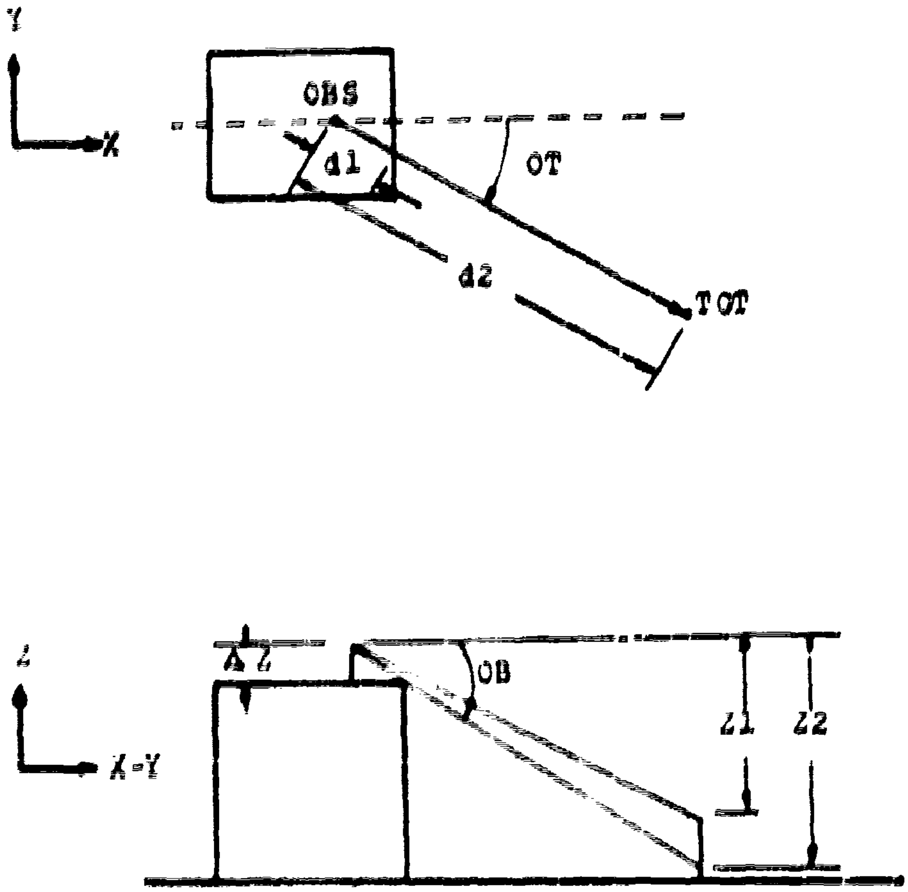


DIAGRAM	PROGRAM
OT	OT.ANGLE = ABS(OBS.TGT.ANGLE)
d1	OBS.BLDG.DIST
d2	OT.XY.DIST
ΔZ	DELTA.Z
OB	OBS.ANGLE.BLDG
Z1	Z.1
Z2	Z.2

PER.CENT.VISIBLE = |Z1-Z2| /(heigth of target)

Figure C-4. Fraction Visible with Observer on the Roof.

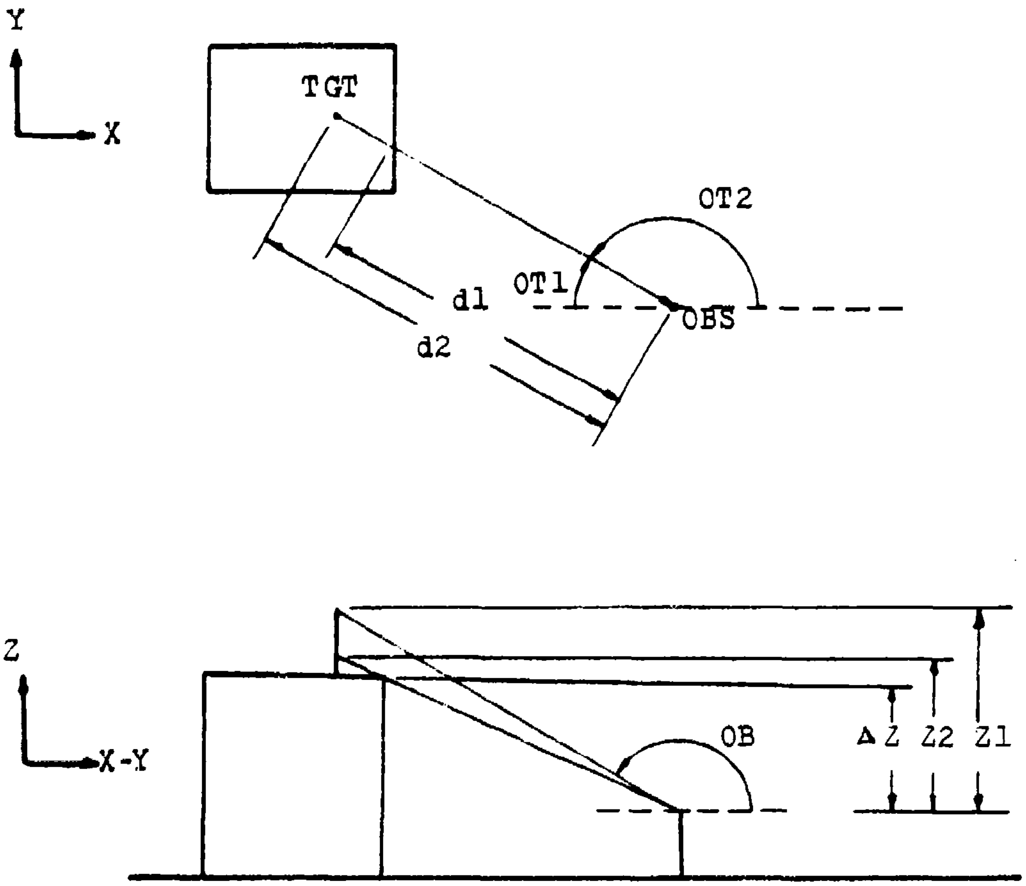


DIAGRAM	PROGRAM
OT1	OBS.TGT.ANGLE
OT2	OT.ANGLE
d1	OBS.BLDG.DIST
d2	OT.XY.DIST
OB	OBS.ANGLE.BLDG
ΔZ	DELTA.Z
Z1	Z.1
Z2	Z.2

PER.CENT.VISIBLE = $|Z1-Z2|$ /(heigth of target)

Figure C-5a. Fraction Visible with Target on the Roof

observer, target and the building and opening being checked. Figure C-4 shows the observer on top of a building, Figure C-5a the observer facing a building. Reference can be made to these figures as the procedure is discussed. The variables discussed are identified on the figure.

The variable DELTA.Z is defined as the difference in elevation between the input parameter defining the elevation of the object be checked and the observer (5). The variable OT.XY.DIST is the distance between the observer and target as measured in the XY-plane (6). The procedure to calculate the fraction visible is done in seven steps:

1. The observer-target angle is redefined as a rotation depending on the side of the building facing the observer. This rotation of the angle allows the program to proceed without having to check whether or not the building being checked is the observer's or not (7-17).

2. The distance the observer is from the leading edge of the building is calculated (19-30). This distance is measured in the XY-plane.

3. The angle between the observer and the leading edge of the building or opening is calculated (31). This angle is measured from the XY-plane to this leading edge.

4. The variable Z.2 is the distance, measured along the z-axis, from the top of the observer to a point on the target intersected by the line from the observer through the leading edge of the building or opening (31).

5. The target can assume a certain posture, thus reducing his effective heigth. The maximum allowable fraction visible is defined based on the particular posture assumed by the target (32-35).

6. The geometrical relationships between the observer, the target and the object being checked, require a check to see if the observer is below the target and above the object being checked (36). This check is made in order to relate the values Z.1 and Z.2 in accordance with this relationship. When the observer is below the target, but above the object being checked, (see Figure C-5b below), when Z.2 is subtracted from Z.1, the fraction visible is less than it should be. To take into consideration this instance, Z.2 is added to Z.1 (39).

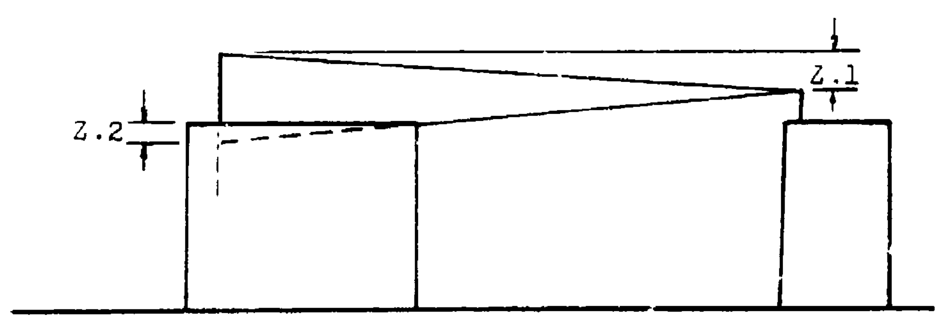


Figure C-5b. Fraction Visible with Target on the Roof

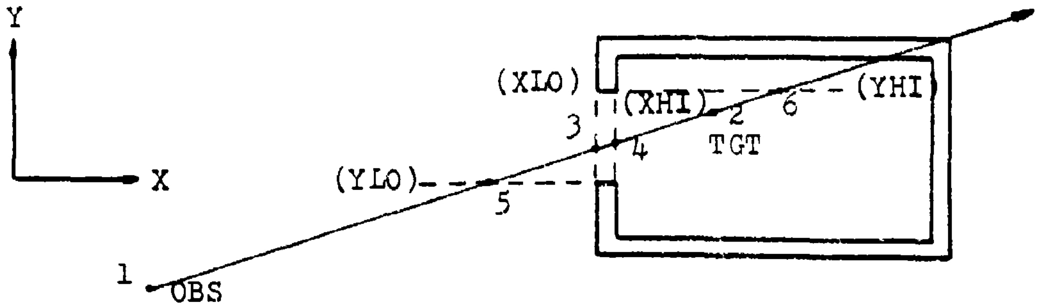
7. The fraction visible is then calculated (37) and the routine is exited.

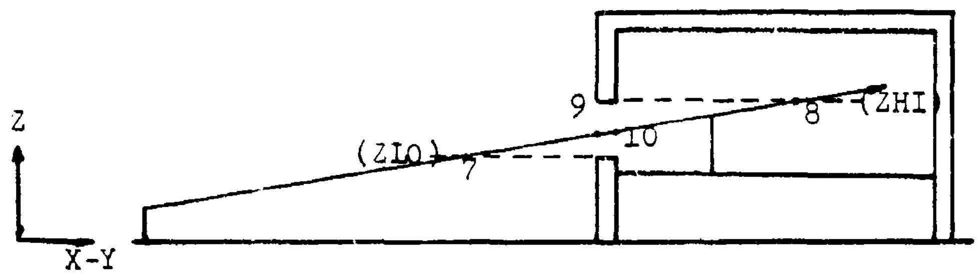
4. Routine TGT.IN.BUILDING

The purpose of this routine is to determine if an observer can see a target through an opening in the target's building. The input variables are the pointers of the target and building, the number of the building, the differences in the coordinates of the observer and target and the side of the building facing the observer. The output variable is a line of sight indicator of either YES of NO. The source code for this routine is attached. References to the code will be made throughout this discussion. Figure C-6 illustrates the location of the observer, target opening and building. All variables defined in this routine are also indicated on the figure.

The principle logic of this routine is exactly the same as in routine CHECK.LOS. The major departure is in the final determination of the existence of line of sight. In CHECK.LOS, if SMIN is less than or equal to SMAX, then line of sight has been broken. In this routine, if SMIN is less than or equal to SMAX, there exists line of sight.

Prior to making any calculations against the planes of the opening, the openings available on the building are screened to insure that only those openings facing the observer are checked (4). The representation of the opening being checked is likened to a rectangular box. The width dimension of the opening is the thickness of the structural material. (The assumption of the model is that structural thickness is uniform throughout the building.)





(1) Numbers 1,3,5 and 9 indicate the sequential movement of S1 along the line of sight. SMIN moves from 1 to 3 to 9.

(2) Numbers 2,4,6 and 8 indicate the sequential movement of S2 along the line of sight. SMAX moves from 2 to 4 to 10.

SMIN < SMAX

THEREFORE

LINE OF SIGHT = YES

Figure C-6. Fraction Visible with Target in the Building

Prior to entering into the portion of the program that checks the X or Y-plane, if the dimension of the opening in the direction being checked is equal to 0.0, then the thickness is added or substracted from that coordinate depending on which direction the opening is facing (12 and 24). The procedure from that point on is similar to CHECK.LOS.

If the planes of the window in all three directions have been broken (36), then the line of sight must have gone through the opening to the target. If this occurs, an indicator of YES is given and the routine is exited. Otherwise, if the target cannot be seen through any of the openings, the routine is exited with NO.

5. Routine UPDATE.TGT.LISTS

This routine has three major functions:

1. To add a detected target into the observer's target list.

2. To remove a target from the observer's target list.

3. To leave the observer's target list along.

The input variables to the routine are the number of the Red and the number of Blue soldier, the line of sight indicator between observer and target and which of the soldiers is the observer. The source code for this routine is attached. References to the code will made throughout this discussion. (See page 141)

The routine is broken into two distinct sections. One section updates the Red observer's target list, the other the Blue's target list. Because the procedures are identical for either observer, only the Red observer will be discussed. Additionally, the procedures to update the list are different depending on the size of the target list, i.e. the number of targets on the list, prior to the update. If the list is empty, the size of the list is one, with the first entry being a zero.

If the size of the target list is equal to one, the following procedure is followed:

1. If the list is empty and there is a detection, add the target to the list (13-16).

2. If the target is a new detection, add the target to the list (17-20).

3. If the observer has lost line of sight to the only target on his list, then the target is removed (22-25).

If there are at least two targets on the observer's list, depending on whether there is or is not line of sight, the flow of the program goes in two directions.

1. If line of sight exists and the target is not on the observer's list, add the new detection to the list (42-45); otherwise do nothing (35-39).

2. If line of sight does not exist and the target is on the target list, remove the detection from the list (50-56); otherwise do nothing.

There are no provisions in this routine to sort these target arrays in order to facilitate the servicing of targets based on some priority of engagement. In this simulation, targets as serviced when the time to enter event KILL.MAN is encountered.

```
**************************************************
* ROUTINE TO CHECK THE LOS BETWEEN AN OBS AND A TGT *
**************************************************

ROUTINE TO CHECK.LOS GIVEN II AND JJ YIELDING K
''RETURNED AS YES.NO
DEFINE II,JJ,AND K AS INTEGER VARIABLES
DEFINE NN.BLKSX AND NN.BLKSY AS INTEGER VARIABLES

''  IF BOTH TARGET AND OBSERVER ARE ON THE SAME FLOOR OF
''  THE SAME HOUSE
''  LINE-OF-SIGHT = YES
IF (MANS.HOUSE(II) EQ MANS.HOUSE(JJ)) AND
   (MANS.FLOOR(II) EQ MANS.FLOOR(JJ))
      LET K=1   LET PER.CENT.VISIBLE=1.0        GO TO 'N3'
''  ELIMINATE PERSONNEL NOT IN THE GRID DESIGNATION
ELSE IF (MANS.BLOCK(II) EQ 0) OR (MANS.BLOCK(JJ) EQ 0)
      GO TO 'N1'
''  MEN IN BUILDINGS BUT NOT AT AN OPENING DO NOT OBSERVE
ELSE IF (MANS.HOUSE(II) NE 0)
      AND (MANS.OPENING(II) EQ 0) GO TO 'N1'
''  OBS-TGT PAIRS IN THE SAME BLDG BUT NOT THE SAME FLOOR
''      ARE ELIMINATED
ELSE IF (MANS.HOUSE(II) EQ MANS.HOUSE(JJ)) AND
           (MANS.FLOOR(II) NE MANS.FLOOR(JJ)) GO TO 'N1'

ELSE
LET X.DIFF=X.MAN(JJ)-X.MAN(II)
IF X.DIFF EQ 0.0 LET X.DIFF=0.01 ALWAYS
LET Y.DIFF=Y.MAN(JJ)-Y.MAN(II)
IF Y.DIFF EQ 0.0 LET Y.DIFF=0.01 ALWAYS
LET Z.DIFF=Z.MAN(JJ)-Z.MAN(II)
IF Z.DIFF EQ 0.0 LET Z.DIFF=.01 ALWAYS
LET NN.BLKSX=1+ABS.F(TRUNC.F(X.DIFF/BLKSIZE))
LET NN.BLKSY=1+ABS.F(TRUNC.F(Y.DIFF/BLKSIZE))
IF NN.BLKSX LT 2  LET NN.BLKSX=2  ALWAYS
IF NN.BLKSY LT 2  LET NN.BLKSY=2  ALWAYS

''  TARGET IN THE OPEN HE IS FAIR GAME FOR LOS CHECK

IF (MANS.OPENING(JJ) EQ 999)    LET K=1      GO TO 'GRID'
ELSE

''  AT THIS POINT THE OBSERVER IS IN AN OPENING, IN THE OPEN
''  OR ON THE ROOF OF HOUSE AND THE TARGT IS IN AN OPENING
''  OR IN A HOUSE.  DEPENDING ON THE OBSERVER'S POSITION,
''  I.E. ON THE ROOF, IN THE OPEN, OR IN AN OPENING, GO TO
''  THE ROUITNE THAT FURTHER DECREASES THE NUMBER OF LOS
```

```
''   CHECKS BY ELIMINATING THOSE PAIRS PROHIBITED LOS BASED
''   ON THEIR LOCATIONS, NOT INTERVENING TERRAIN.

     IF (MANS.OPENING(II) NE 999)        GO TO 'C1'
     ELSE NOW CHECK.OPEN.OBS GIVEN X.DIFF,Y.DIFF AND JJ
          YIELDING K
     GO TO 'C2'

     'C1'
     IF (MANS.DIR(II) EQ 1) OR (MANS.DIR(II) EQ 3)
          NOW CHECK.NORTH.SOUTH GIVEN X.DIFF,Y.DIFF,II AND JJ
               YIELDING K
     ELSE NOW CHECK.EAST.WEST GIVEN X.DIFF,Y.DIFF,II AND JJ
          YIELDING K

     'C2'
     ALWAYS IF K EQ 1 GO TO 'GRID'  ELSE GO TO 'N2'

''   NOW THE TGT AND OBS ARE READY FOR A LINE OF SIGHT CHECK.
''   NEXT MINIMIZE THE NUMBER OF GRID SQUARES ONE MUST CHECK
''   ALONG THE OBSERVER-TARGET LINE.

     'GRID'
     DEFINE NGRSQ,ISGX,ISGY,IX AND IY AS INTEGER VARIABLES
     DEFINE IGX AND IGY AS 1-DIMENSIONAL INTEGER ARRAYS
     RESERVE IGX(*) AS NN.BLKSX
     RESERVE IGY(*) AS NN.BLKSY
     LET NGRSQ=0
     IF X.DIFF GT 0.0  LET ISGX=-1  LET XINC=BLKSIZE/X.DIFF
     JUMP AHEAD
     ELSE LET ISGX =1  LET XINC=-BLKSIZE/X.DIFF

     HERE
     IF Y.DIFF GT 0.0  LET ISGY=-1  LET YINC=BLKSIZE/Y.DIFF
     JUMP AHEAD
     ELSE LET ISGY=1  LET YINC=-BLKSIZE/Y.DIFF

     HERE
     LET IX=1+TRUNC.F(X.MAN(JJ)/BLKSIZE)
     LET IY=1+TRUNC.F(Y.MAN(JJ)/BLKSIZE)
     IF IX LT 1 LET IX=1 ALWAYS
     IF IY LT 1 LET IY=1 ALWAYS
     IF IX GT NBLKX LET IX=NBLKX ALWAYS
     IF IY GT NBLKY LET IY=NBLKY ALWAYS
     LET XSTEP=(X.MAN(JJ)-BLKSIZE*(IX+.5*(ISGX-1.)))/X.DIFF
     LET YSTEP=(Y.MAN(JJ)-BLKSIZE*(IY+.5*(ISGY-1.)))/Y.DIFF

     'FIND'
```

```
ADD 1 TO NGRSQ
LET IGX(NGRSQ)=IX    LET IGY(NGRSQ)=IY
IF (XSTEP LE 1.0) OR (YSTEP LE 1.0)
     IF XSTEP LT YSTEP ADD ISGX TO IX  ADD XINC TO XSTEP
                        GO TO 'FIND'
     ELSE IF XSTEP GT YSTEP JUMP AHEAD
     ELSE ADD ISGX TO IX   ADD XINC TO XSTEP
     HERE ADD ISGY TO IY   ADD YINC TO YSTEP
     GO TO 'FIND'
ALWAYS

THE NUMBER OF BLOCKS TO BE CHECKED HAS BEEN MINIMIZED.
CHECK THE HOUSES IN THOSE BLOCKS FOR INTERFERENCE,
BY SEEING IF THREE PLANES OF THE HOUSE INTERSECT THE
OBSERVER-TARGET LINE.

TO MINIMIZE THE CALCULATIONS, AN INTEGER VARIABLE, IFLAG
IS USED AND WILL INDICATE WHEN AN INTERVENING TERRAIN
FEATURE IS ON THE OBS-TGT LINE-OF-SIGHT LINE; THEREFORE
NOT ALL BUILDINGS WILL BE CHECKED FOR PER.CENT.VISIBLE.

DEFINE IHOUSE,IFLAG AND FACE AS INTEGER VARIABLES
LET X.DIFF=-X.DIFF LET Y.DIFF=-Y.DIFF LET Z.DIFF=-Z.DIFF
LET PER.CENT.VISIBLE = 1.0
FOR J=1 TO NGRSQ, DO
     FOR IHOUSE=1 TO NN.HOUSE, WITH
         (BLK.GRID(IGX(J),IGY(J))
               EQ BLOCK(PT.HOUSE(IHOUSE))), DO

     LET IFLAG=0
     LET SMIN=0.0  LET SMAX=1.0
     LET SLO=(X.1(PT.HOUSE(IHOUSE))-X.MAN(JJ))/X.DIFF
     LET SHI=(X.2(PT.HOUSE(IHOUSE))-X.MAN(JJ))/X.DIFF
     LET S1=MIN.F(SLO,SHI)    LET S2=MAX.F(SLO,SHI)
     LET SMIN=MAX.F(SMIN,S1)   LET SMAX=MIN.F(SMAX,S2)
     IF SMIN-SMAX LT 0.0      ADD 1 TO IFLAG    ALWAYS

     LET SLO=(Y.1(PT.HOUSE(IHOUSE))-Y.MAN(JJ))/Y.DIFF
     LET SHI=(Y.4(PT.HOUSE(IHOUSE))-Y.MAN(JJ))/Y.DIFF
     LET S1=MIN.F(SLO,SHI)    LET S2=MAX.F(SLO,SHI)
     LET SMIN=MAX.F(SMIN,S1)   LET SMAX=MIN.F(SMAX,S2)
     IF SMIN-SMAX LT 0.0      ADD 1 TO IFLAG    ALWAYS

     LET SLO=-Z.MAN(JJ)/Z.DIFF
     LET SHI=(DIM.Z(PT.HOUSE(IHOUSE))-Z.MAN(JJ))/Z.DIFF
     LET S1=MIN.F(SLO,SHI)    LET S2=MAX.F(SLO,SHI)
     LET SMIN=MAX.F(SMIN,S1)   LET SMAX=MIN.F(SMAX,S2)
     IF SMIN-SMAX GT 0.0      ADD 1 TO IFLAG    ALWAYS
```

```
        IF SMIN LE SMAX      GO TO 'N4'      ''LOS IS BLOCKED
        ELSE IF IFLAG NE 3  GO TO 'NO' ''BLDG IS NOT ON LOS

        ELSE
        NOW FIND.FACE GIVEN II,JJ,X.DIFF,Y.DIFF,Z.DIFF,
             PT.HOUSE(IHOUSE) YIELDING FACE

        LET Z1=DIM.Z(PT.HOUSE(IHOUSE))
        CALL PERCENT.VISIBLE GIVEN II,JJ,PT.HOUSE(IHOUSE),
             X.DIFF,Y.DIFF,Z.DIFF,Z1,FACE  YIELDING FCN.VIS

        LET PER.CENT.VISIBLE=MIN.F(PER.CENT.VISIBLE,FCN.VIS)

        GO TO 'NO'

'N4' LET K=0
             IF (NAME(PT.HOUSE(IHOUSE)) EQ MANS.HOUSE(JJ))
             AND (MANS.OPENING(JJ) EQ 0)
             NOW FIND.FACE GIVEN II,JJ,X.DIFF,Y.DIFF,Z.DIFF,
                 PT.HOUSE(IHOUSE)      YIELDING FACE
             CALL TGT.IN.BUILDING GIVEN JJ,PT.HOUSE(IHOUSE),IHOUSE,
                  X.DIFF,Y.DIFF,Z.DIFF,FACE      YIELDING K,Z3
             ALWAYS IF K EQ 0  GO TO 'N2'    ELSE

             CALL PERCENT.VISIBLE GIVEN II,JJ,PT.HOUSE(IHOUSE),
                  X.DIFF,Y.DIFF,Z.DIFF,Z3,FACE        YIELDING FCN.VIS
             LET PER.CENT.VISIBLE=MIN.F(PER.CENT.VISIBLE,FCN.VIS)

'NO'         LET K=1
        LOOP
LOOP

'N3'
LET RANGE.TO.TARGET=SQRT.F((X.DIFF**2)+(Y.DIFF**2)+(Z.DIFF**2))
GO TO 'N2'

'N1' LET K=0

'N2' END''OF ROUTINE TO SEE IF THERE EXISTS LINE-OF-SIGHT
```

```
*********************************************************
* ROUTINE TO ELIMINATE O-T PAIRS WITH OBS IN THE OPEN *
*********************************************************

  ROUTINE TO CHECK.OPEN.OBS GIVEN X.DIFF,Y.DIFF,JTGT YIELDING KLOS
                                                     ''RETURNED TO K
  DEFINE JTGT,KLOS AS INTEGER VARIABLES

  IF (MANS.DIR(JTGT) NE 0) JUMP AHEAD     ELSE GO TO 'YES'

HERE      IF Y.DIFF GT 0 JUMP AHEAD
  ELSE IF X.DIFF GT 0.0  GO TO 'C1'
          ELSE IF (MANS.DIR(JTGT) EQ 3) OR (MANS.DIR(JTGT) EQ 2)
               GO TO 'YES'    ELSE GO TO 'NO'

'C1'IF (MANS.DIR(JTGT) EQ 3) OR (MANS.DIR(JTGT) EQ 4)
          GO TO 'YES'    ELSE GO TO 'NO'

HERE      IF X.DIFF GT 0.0  GO TO 'C2'
          ELSE IF (MANS.DIR(JTGT) EQ 1) OR (MANS.DIR(JTGT) EQ 2)
               GO TO 'YES'    ELSE GO TO 'NO'

'C2'IF (MANS.DIR(JTGT) EQ 1) OR (MANS.DIR(JTGT) EQ 4)
          GO TO 'YES'    ELSE GO TO 'NO'

'YES'     LET KLOS=1     GO TO 'END'
'NO'      LET KLOS=0

'END'END''OF ROUTINE TO CHECK OBS-TGT LOS FOR OBSERVER IN THE OPEN
```

```
*****************************************************************
* ROUTINE TO ELIMINATE O-T PAIRS WITH OBS FACING NORTH OR SOUTH *
*****************************************************************

   ROUTINE TO CHECK.NORTH.SOUTH GIVEN X.DIFF,Y.DIFF,IOBS,JTGT YIELDING
   KLOS''RETURNED TO K
   DEFINE IOBS,JTGT,KLOS AS INTEGER VARIABLES

   IF Y.DIFF LT 0.0 GO TO 'H2'
        ELSE IF (MANS.DIR(IOBS) EQ 3) AND (MANS.DIR(JTGT) EQ 0)
             GO TO 'YES'
        ELSE IF (MANS.DIR(IOBS) EQ 3) AND (MANS.DIR(JTGT) EQ 1)
             GO TO 'YES'
        ELSE IF X.DIFF GT 0 GO TO 'C3'
             ELSE IF (MANS.DIR(IOBS) EQ 3) AND (MANS.DIR(JTGT) EQ 2)
                  GO TO 'YES'    ELSE GO TO 'NO'

'C3'    IF (MANS.DIR(IOBS) EQ 3) AND (MANS.DIR(JTGT) EQ 4)
             GO TO 'YES'    ELSE GO TO 'NO'

'H2'    IF (MANS.DIR(IOBS) EQ 1) AND (MANS.DIR(JTGT) EQ 0)
             GO TO 'YES'
        ELSE IF (MANS.DIR(IOBS) EQ 1) AND (MANS.DIR(JTGT) EQ 3)
             GO TO 'YES'
        ELSE IF X.DIFF LT 0 GO TO 'C4'
             ELSE IF (MANS.DIR(IOBS) EQ 1) AND (MANS.DIR(JTGT) EQ 4)
                  GO TO 'YES'    ELSE GO TO 'NO'

'C4'    IF (MANS.DIR(IOBS) EQ 1) AND (MANS.DIR(JTGT) EQ 2)
             GO TO 'YES'    ELSE GO TO 'NO'

'YES'   LET KLOS=1     GO TO 'END'
'NO'    LET KLOS=0

'END'END''OF ROUTINE TO CHECK OBS-TGT LOS WITH N-S OBSERVER
```

```
*****************************************************************
* ROUTINE TO ELIMINATE O-T PAIRS WITH OBSERVER FACING EAST OR WEST *
*****************************************************************

  ROUTINE TO CHECK.EAST.WEST GIVEN X.DIFF,Y.DIFF,IOBS,JTGT YIELDING
  KLOS''RETURNED TO K
  DEFINE IOBS,JTGT,KLOS AS INTEGER VARIABLES

  IF X.DIFF LT 0.0 GO TO 'H1'
       ELSE IF (MANS.DIR(IOBS) EQ 2) AND (MANS.DIR(JTGT) EQ 0)
            GO TO 'YES'
       ELSE IF (MANS.DIR(IOBS) EQ 2) AND (MANS.DIR(JTGT) EQ 4)
            GO TO 'YES'
       ELSE IF Y.DIFF GT 0 GO TO 'C1'
            ELSE IF (MANS.DIR(IOBS) EQ 2) AND (MANS.DIR(JTGT) EQ 3)
                 GO TO 'YES'     ELSE GO TO 'NO'

'C1'     IF (MANS.DIR(IOBS) EQ 2) AND (MANS.DIR(JTGT) EQ 1)
              GO TO 'YES'    ELSE GO TO 'NO'

'H1'     IF (MANS.DIR(IOBS) EQ 4) AND (MANS.DIR(JTGT) EQ 0)
            GO TO 'YES'
         ELSE IF (MANS.DIR(IOBS) EQ 4) AND (MANS.DIR(JTGT) EQ 2)
            GO TO 'YES'
         ELSE IF Y.DIFF LT 0 GO TO 'C2'
            ELSE IF (MANS.DIR(IOBS) EQ 4) AND (MANS.DIR(JTGT) EQ 1)
                 GO TO 'YES'     ELSE GO TO 'NO'

'C2'     IF (MANS.DIR(IOBS) EQ 4) AND (MANS.DIR(JTGT) EQ 3)
            GO TO 'YES'    ELSE GO TO 'NO'

'YES'    LET KLOS=1      GO TO 'END'
'NO'     LET KLOS=0

'END'END''OF ROUTINE TO CHECK OBS-TGT LINE WITH E-W OBSERVER
PAGE
```

```
*********************************************************
* ROUTINE TO FIND THE FACE OF BUILDING FACING THE OBSERVER *
*********************************************************

ROUTINE TO FIND.FACE GIVEN IOBS,JTGT,X,Y,Z,PTHSE YIELDING FACE.OBS

DEFINE IOBS,JTGT,PTHSE,FACE.OBS AS INTEGER VARIABLES

'' OBS-TGT ANGLE AS MEASURED IN THE X-Y PLANE
LET OBS.TGT.ANGLE=ARCTAN.F(-Y,-X)

'' INITIATING THE STEPS TO DETERMINE WHICH FACE OF THE BUILDING THE
'' OBSERVER IS FACING.  THE PURPOSE IS TWOFOLD:  (1)  TO REDUCE THE
'' NUMBER OF OPENINGS TO BE CHECKED IF THE TARGET IS INSIDE THE
'' BUILDING AND (2)  TO ASSIST IN THE CALCULATION OF THE FRACTION OF
'' THE TARGET VISIBLE TO THE OBSERVER.

'' NOTE:  THE OBSERVERS BUILDING HAS TC BE HANDLED SEPARATELY

IF MANS.HOUSE(IOBS) NE NAME(PTHSE)      JUMP AHEAD      ELSE
LET A1=ARCTAN.F(Y.1(PTHSE)-Y.MAN(IOBS),X.1(PTHSE)-X.MAN(IOBS))
     IF (A1 LT PI.C+.000001) AND (A1 GT PI.C-.000001)
          LET A1 = -A1             ALWAYS
LET A2=ARCTAN.F(Y.2(PTHSE)-Y.MAN(IOBS),X.2(PTHSE)-X.MAN(IOBS))
LET A3=ARCTAN.F(Y.3(PTHSE)-Y.MAN(IOBS),X.3(PTHSE)-X.MAN(IOBS))
LET A4=ARCTAN.F(Y.4(PTHSE)-Y.MAN(IOBS),X.4(PTHSE)-X.MAN(IOBS))

IF (OBS.TGT.ANGLE LT A2) AND (OBS.TGT.ANGLE GT A1)
     LET FACE.OBS=1        GO TO 'END'     ELSE
IF (OBS.TGT.ANGLE LT A3) AND (OBS.TGT.ANGLE GT A2)
     LET FACE.OBS=2        GO TO 'END'     ELSE
IF (OBS.TGT.ANGLE LT A4) AND (OBS.TGT.ANGLE GT A3)
     LET FACE.OBS=3        GO TO 'END'     ELSE
     LET FACE.OBS=4        GO TO 'END'

'' ALL BUILDINGS OTHER THAN THE OBSERVERS ARE CHECKED HERE

HERE
IF (X.MAN(IOBS) GE X.1(PTHSE)) AND (X.MAN(IOBS) LE X.2(PTHSE))
     GO TO 'CHECK.Y'  ELSE
IF (Y.MAN(IOBS) GE Y.1(PTHSE)) AND (Y.MAN(IOBS) LE Y.4(PTHSE))
     GO TO 'CHECK.X'  ELSE
IF (X.MAN(IOBS) LT X.1(PTHSE)) AND (Y.MAN(IOBS) LT Y.1(PTHSE))
     GO TO 'CHECK.ONE.FOUR'  ELSE
IF (X.MAN(IOBS) LT X.1(PTHSE)) AND (Y.MAN(IOBS) GT Y.4(PTHSE))
     GO TO 'CHECK.FOUR.THREE'  ELSE
IF (X.MAN(IOBS) GT X.2(PTHSE)) AND (Y.MAN(IOBS) LT Y.1(PTHSE))
     GO TO 'CHECK.TWO.ONE'  ELSE
```

```
IF (X.MAN(IOBS) GT X.2(PTHSE)) AND (Y.MAN(IOBS) GT Y.4(PTHSE))
          GO TO 'CHECK.THREE.TWO'
ELSE GO TO 'END'     ''SOMETHING HAS GONE WRONG

'' THIS IS THE POINT THAT ISOLATES THE SPECIFIC FACE OF THE BUILDING
'' FACING THE OBSERVER.

'CHECK.Y'
IF (Y.MAN(IOBS) LT Y.1(PTHSE))  LET FACE.OBS=1     GO TO 'END'
              ELSE LET FACE.OBS=3       GO TO 'END'
'CHECK.X'
IF (X.MAN(IOBS) LT X.1(PTHSE))  LET FACE.OBS=4     GO TO 'END'
              ELSE LET FACE.OBS=2       GO TO 'END'
'CHECK.ONE.FOUR'
IF(OBS.TGT.ANGLE LT ARCTAN.F(Y.1(PTHSE)-Y.MAN(IOBS),
                                  X.1(PTHSE)-X.MAN(IOBS)))
       LET FACE.OBS=1       GO TO 'END'
       ELSE LET FACE.OBS=4       GO TO 'END'
'CHECK.FOUR.THREE'
IF(OBS.TGT.ANGLE LT ARCTAN.F(Y.4(PTHSE)-Y.MAN(IOBS),
                                  X.1(PTHSE)-X.MAN(IOBS)))
       LET FACE.OBS=4        GO TO 'END'
       ELSE LET FACE.OBS=3        GO TO 'END'
'CHECK.THREE.TWO'
IF(OBS.TGT.ANGLE LT ARCTAN.F(Y.4(PTHSE)-Y.MAN(IOBS),
                                  X.2(PTHSE)-X.MAN(IOBS)))
       LET FACE.OBS=3        GO TO 'END'
       ELSE LET FACE.OBS=2        GO TO 'END'
'CHECK.TWO.ONE'
IF(OBS.TGT.ANGLE LT ARCTAN.F(Y.1(PTHSE)-Y.MAN(IOBS),
                                  X.2(PTHSE)-X.MAN(IOBS)))
       LET FACE.OBS=2        GO TO 'END'
       ELSE LET FACE.OBS=1

'END' END''OF ROUTINE TO FIND THE SIDE OF BLDG FACING THE OBSERVER
```

```
*********************************************
* CALCULATE THE FRACTION VISIBLE OF THE TARGET *
*********************************************

ROUTINE FOR PERCENT.VISIBLE GIVEN IOBS,JTGT,PTHSE,X,Y,Z,Z2,FACE.OBS
     YIELDING PCT.VIS

Z2 = EITHER (1) ELEVATION OF THE TGTS BLDG OR
            (2) ELEVATION OF THE BOTTOM OF THE OPENING THROUGH
                WHICH THE OBS IS LOOKING AT THE TGT

DEFINE IOBS,JTGT,PTHSE,FACE.OBS AS INTEGER VARIABLES

LET HALF.PI=PI.C/2.0
LET DELTA.Z=Z2-Z.MAN(IOBS)
LET OT.XY.DIST=SQRT.F((X**2)+(Y**2))''O-T DIST IN XY PLANE

IF (FACE.OBS EQ 1) OR (FACE.OBS EQ 3)         JUMP AHEAD      ELSE
IF (OBS.TGT.ANGLE LT -HALF.PI) AND (OBS.TGT.ANGLE GT -PI.C)
LET OT.ANGLE = PI.C+OBS.TGT.ANGLE              GO TO 'NEXT'    ELSE
IF (OBS.TGT.ANGLE GT HALF.PI) AND (OBS.TGT.ANGLE LT PI.C)
LET OT.ANGLE = PI.C-OBS.TGT.ANGLE              GO TO 'NEXT'    ELSE
LET OT.ANGLE = ABS.F(OBS.TGT.ANGLE)            GO TO 'NEXT'

HERE      IF (OBS.TGT.ANGLE LT 0.0) AND (OBS.TGT.ANGLE GT -PI.C)
LET OT.ANGLE = ABS.F(HALF.PI+OBS.TGT.ANGLE)    GO TO 'NEXT'    ELSE
IF (OBS.TGT.ANGLE LT HALF.PI) AND (OBS.TGT.ANGLE GT 0.0)
LET OT.ANGLE = HALF.PI-OBS.TGT.ANGLE           GO TO 'NEXT'    ELSE
LET OT.ANGLE = OBS.TGT.ANGLE-HALF.PI

'NEXT'
GO TO ONE,TWO,THREE,FOUR PER FACE.OBS

'ONE'
LET OBS.BLDG.DIST=(Y.MAN(IOBS)-Y.1(PTHSE))/COS.F(OT.ANGLE)
     JUMP AHEAD
'TWO'
LET OBS.BLDG.DIST=(X.MAN(IOBS)-X.2(PTHSE))/COS.F(OT.ANGLE)
     JUMP AHEAD
'THREE'
LET OBS.BLDG.DIST=(Y.MAN(IOBS)-Y.4(PTHSE))/COS.F(OT.ANGLE)
     JUMP AHEAD
'FOUR'
LET OBS.BLDG.DIST=(X.MAN(IOBS)-X.1(PTHSE))/COS.F(OT.ANGLE)

HERE      LET OBS.ANGLE.BLDG=ARCTAN.F(DELTA.Z,OBS.BLDG.DIST)

LET Z.1=ABS.F(Z)     LET Z.2=ABS.F(OT.XY.DIST*TAN.F(OBS.ANGLE.BLDG))
```

```
''  PCT.VIS = FRACTION OF THE TARGET VISIBLE TO THE OBSERVER BASED
''  ON THE TARGETS HEIGTH BEING 2.0 METERS
''  MAX.PCT.VIS = MAXIMUM ALLOWABLE PCT.VIS

IF POSTURE(JTGT) EQ 1     LET MAX.PCT.VIS = 1.0    JUMP AHEAD ELSE
IF POSTURE(JTGT) EQ 2     LET MAX.PCT.VIS =  .75   JUMP AHEAD ELSE
IF POSTURE(JTGT) EQ 3     LET MAX.PCT.VIS =  .50   JUMP AHEAD ELSE
IF POSTURE(JTGT) EQ 4     LET MAX.PCT.VIS =  .25   ALWAYS

HERE IF Z.MAN(IOBS) LE Z.MAN(JTGT) AND Z.MAN(ICBS) GE Z2  JUMP AHEAD
ELSE       LET PCT.VIS = MIN.F(ABS.F(Z.1-Z.2)/2.0,MAX.PCT.VIS)
GO TO 'END'

HERE       LET PCT.VIS = MIN.F(ABS.F(Z.1+Z.2)/2.0,MAX.PCT.VIS)

'END'
END''OF ROUTINE TO CALCULATE FRACTICN VISIBLE OF A TARGET
```

```
   ****************************************************************
'' * ROUTINE TO CHECK LOS THROUGH THE OPENINGS OF A TARGETS BUILDING *
   ****************************************************************

   ROUTINE FOR TGT.IN.BUILDING GIVEN JTGT,PTHSE,IHSE,X,Y,Z,FACE.OBS
         YIELDING KLOS AND Z1

'' LOGIC FOR LINE-OF-SIGHT IS OPPOSITE OF THAT CONTAINED IN ROUTINE
'' CHECK.LOS, I.E.  BY GIVING THE OPENING DIMENSIONS IN THE X, Y, AND
'' Z-PLANES, LINE-OF-SIGHT EXISTS WHEN THE OBS-TGT LINE INTERSECTS THE
'' OPENING "BOX".

   DEFINE JTGT,PTHSE,IHSE,FACE.OBS,KLOS,I,JOPEN AS INTEGER VARIABLES

   FOR I=1 TO N.OPENINGS(PTHSE) WITH (OPEN.FACE(IHSE,I) EQ FACE.OBS),DO
         LET JOPEN = 3*(I-1) + 1
         LET SMIN=0.0    LET SMAX=1.0    LET O.D=0.0

         IF OPEN.DIMEN(IHSE,JOPEN) GT 0.0    JUMP AHEAD ELSE
         IF FACE.OBS EQ 4  LET O.D=THICK(PTHSE)   JUMP AHEAD
         ELSE    LET O.D=-THICK(PTHSE)

         HERE
         LET  TEMPX1=OPENING.ARRAY(IHSE,JOPEN)+X.1(PTHSE)
         LET  TEMPX2=TEMPX1+OPEN.DIMEN(IHSE,JOPEN)+O.D
         LET  X1=MIN.F(TEMPX1,TEMPX2)    LET X2=MAX.F(TEMPX1,TEMPX2)
         LET  SLO=(X1-X.MAN(JTGT))/X
         LET  SHI=(X2-X.MAN(JTGT))/X
         LET  S1=MIN.F(SLO,SHI)     LET  S2=MAX.F(SLO,SHI)
         LET  SMIN=MAX.F(SMIN,S1)  LET  SMAX=MIN.F(SMAX,S2)

         LET O.D=0.0
         IF OPEN.DIMEN(IHSE,JOPEN+1) GT 0.0      JUMP AHEAD ELSE
         IF FACE.OBS EQ 1 LET O.D=THICK(PTHSE)  JUMP AHEAD
         ELSE    LET O.D=-THICK(PTHSE)

         HERE
         LET  TEMPY1=OPENING.ARRAY(IHSE,JOPEN+1)+Y.1(PTHSE)
         LET  TEMPY4=TEMPY1+OPEN.DIMEN(IHSE,JOPEN+1)+O.D
         LET  Y1=MIN.F(TEMPY1,TEMPY4)    LET Y4=MAX.F(TEMPY1,TEMPY4)
         LET  SLO=(Y1-Y.MAN(JTGT))/Y
         LET  SHI=(Y4-Y.MAN(JTGT))/Y
         LET  S1=MIN.F(SLO,SHI)     LET  S2=MAX.F(SLO,SHI)
         LET  SMIN=MAX.F(SMIN,S1)  LET  SMAX=MIN.F(SMAX,S2)

         LET  Z1=OPENING.ARRAY(IHSE,JOPEN+2)
         LET  Z2=Z1+OPEN.DIMEN(IHSE,JOPEN+2)
         LET  SLO=(Z1-Z.MAN(JTGT))/Z
```

```
      LET SHI=(Z2-Z.MAN(JTGT))/Z
      LET S1=MIN.F(SLO,SHI)     LET S2=MAX.F(SLO,SHI)
      LET SMIN=MAX.F(SMIN,S1)   LET SMAX=MIN.F(SMAX,S2)

      IF SMIN LE SMAX     LET KLOS=1          GO TO 'PERCENT'
      ELSE LOOP

LET KLOS=0     LET Z1=0.0

'PERCENT' END''OF ROUTINE TO CHECK LOS FOR TGT IN A BUILDING
```

```
*********************************
* ROUTINE TO UPDATE TARGET LISTS *
*********************************

ROUTINE TO UPDATE.TGT.LISTS GIVEN RED,BLUE,Y.N AND WHO.OBS
DEFINE RED,BLUE AND WHO.OBS AS INTEGER VARIABLES
DEFINE Y.N AND KTEMP AS INTEGER VARIABLES
DEFINE K,SIZE.RED AND SIZE.BLUE AS INTEGER VARIABLES
DEFINE TEMP.RED.LIST AND TEMP.BLUE.LIST AS INTEGER 1-DIMENSIONAL
      ARRAYS

GO TO ONE,TWO PER WHO.OBS

'ONE' IF ACTIVITY(PT.MAN(RED)) NE 0      JUMP AHEAD
      ELSE     GO TO 'FINIS'
HERE        LET RED.TGT.LIST(*)=PERS.RED.LIST(1,RED)
            LET SIZE.RED=DIM.F(RED.TGT.LIST(*))
IF SIZE.RED GT 1 GO TO 'ADDBLUE'
LIST IS EMPTY AND DETECTION=YES
      ELSE IF ((RED.TGT.LIST(1) EQ 0) AND (Y.N EQ 1))
            LET RED.TGT.LIST(1)=PT.MAN(BLUE+NN.RED)
            LET PERS.RED.LIST(1,RED)=RED.TGT.LIST(*)
            LET RED.TGT.LIST(*)=0  GO TO 'END'
MAN ON THE LIST IS NOT THE CURRENT DETECTION
      ELSE IF (RED.TGT.LIST(1) NE PT.MAN(BLUE+NN.RED)) AND (Y.N EQ 1)
            RESERVE TEMP.RED.LIST(*) AS 2
            LET TEMP.RED.LIST(1)=RED.TGT.LIST(1)
            LET TEMP.RED.LIST(2)=PT.MAN(BLUE+NN.RED)
            GO TO 'ZERO'
OBS HAS LOST LOS TO MAN ALREADY ON HIS LIST
      ELSE IF (RED.TGT.LIST(1) EQ PT.MAN(BLUE+NN.RED)) AND (Y.N EQ 0)
            LET RED.TGT.LIST(1)=0
            LET PERS.RED.LIST(1,RED)=RED.TGT.LIST(*)
            LET RED.TGT.LIST(*)=0     ALWAYS GO TO 'END'
'ZERO'
LET PERS.RED.LIST(1,RED)=TEMP.RED.LIST(*)
LET TEMP.RED.LIST(*)=0
LET RED.TGT.LIST(*)=0  GO TO 'END'

'ADDBLUE'
IF Y.N EQ 0 GO TO 'REMOVE1'  ELSE''ADD BLUE SOLDIER TO LIST
      RESERVE TEMP.RED.LIST(*) AS SIZE.RED+1

      LET K=1
'INC1
IF PT.MAN(BLUE+NN.RED) NE RED.TGT.LIST(K)    JUMP AHEAD
      ELSE LET PERS.RED.LIST(1,RED)=RED.TGT.LIST(*)
            LET TEMP.RED.LIST(*)=0
```

```
            LET RED.TGT.LIST(*)=0
            GO TO 'END'
HERE LET TEMP.RED.LIST(K)=RED.TGT.LIST(K)     ADD 1 TO K
     IF K LE SIZE.RED    GO TO 'INC1'    ELSE

     LET TEMP.RED.LIST(SIZE.RED+1)=PT.MAN(BLUE+NN.RED)
     LET PERS.RED.LIST(1,RED)=TEMP.RED.LIST(*)
     LET TEMP.RED.LIST(*)=0
     LET RED.TGT.LIST(*)=0
     GO TO 'END'

'REMOVE1'
RESERVE TEMP.RED.LIST(*) AS SIZE.RED-1
LET KTEMP=0
FOR K=1 TO SIZE.RED WITH PT.MAN(BLUE+NN.RED) NE RED.TGT.LIST(K), DO
     ADD 1 TO KTEMP
     LET TEMP.RED.LIST(KTEMP)=RED.TGT.LIST(K)
LOOP
LET PERS.RED.LIST(1,RED)=TEMP.RED.LIST(*)
LET TEMP.RED.LIST(*)=0
LET RED.TGT.LIST(*)=0
GO TO 'END'

'TWO'
IF ACTIVITY(PT.MAN(BLUE+NN.RED)) NE 0      JUMP AHEAD      ELSE
     GO TO 'FINIS'
HERE LET BLUE.TGT.LIST(*)=PERS.BLUE.LIST(1,BLUE)
     LET SIZE.BLUE=DIM.F(BLUE.TGT.LIST(*))
     IF SIZE.BLUE GT 1 GO TO 'ADDRED'
     ELSE IF ((BLUE.TGT.LIST(1) EQ 0) AND (Y.N EQ 1))
            LET BLUE.TGT.LIST(1)=PT.MAN(RED)
            LET PERS.BLUE.LIST(1,BLUE)=BLUE.TGT.LIST(*)
            LET BLUE.TGT.LIST(*)=0   GO TO 'END'
     ELSE IF (BLUE.TGT.LIST(1) NE PT.MAN(RED)) AND (Y.N EQ 1)
            RESERVE TEMP.BLUE.LIST(*) AS 2
            LET TEMP.BLUE.LIST(1)=BLUE.TGT.LIST(1)
            LET TEMP.BLUE.LIST(2)=PT.MAN(RED) GO TO 'STAR'
     ELSE IF (BLUE.TGT.LIST(1) EQ PT.MAN(RED)) AND (Y.N EQ 0)
            LET BLUE.TGT.LIST(1)=0
            LET PERS.BLUE.LIST(1,BLUE)=BLUE.TGT.LIST(*)
            LET BLUE.TGT.LIST(*)=0   ALWAYS GO TO 'END'
'STAR'
LET PERS.BLUE.LIST(1,BLUE)=TEMP.BLUE.LIST(*)
LET TEMP.BLUE.LIST(*)=0
LET BLUE.TGT.LIST(*)=0      GO TO 'END'

'ADDRED'
IF Y.N EQ 0 GO TO 'REMOVE2'  ELSE''ADD RED SOLDIER TO LIST
```

```
RESERVE TEMP.BLUE.LIST(*) AS SIZE.BLUE+1

        LET K=1

'INC1'
IF PT.MAN(RED) NE BLUE.TGT.LIST(K)  JUMP AHEAD
        ELSE LET PERS.BLUE.LIST(1,BLUE)=BLUE.TGT.LIST(*)
                LET TEMP.BLUE.LIST(*)=0
                LET BLUE.TGT.LIST(*)=0
                GO TO 'END'

HERE LET TEMP.BLUE.LIST(K)=BLUE.TGT.LIST(K)          ADD 1 TO K
        IF K LE SIZE.BLUE  GO TO 'INC2'  ELSE

        LET TEMP.BLUE.LIST(SIZE.BLUE+1)=PT.MAN(RED)
        LET PERS.BLUE.LIST(1,BLUE)=TEMP.BLUE.LIST(*)
        LET TEMP.BLUE.LIST(*)=0
        LET BLUE.TGT.LIST(*)=0
        GO TO 'END'

'REMOVE2'
RESERVE TEMP.BLUE.LIST(*) AS SIZE.BLUE-1
LET KTEMP=0
FOR K=1 TO SIZE.BLUE
        WITH PT.MAN(RED) NE BLUE.TGT.LIST(K), DO
        ADD 1 TO KTEMP
        LET TEMP.BLUE.LIST(KTEMP)=BLUE.TGT.LIST(K)
LOOP
LET PERS.BLUE.LIST(1,BLUE)=TEMP.BLUE.LIST(*)
LET TEMP.BLUE.LIST(*)=0
LET BLUE.TGT.LIST(*)=0

'END'IF WHO.OBS EQ 1      JUMP AHEAD      ELSE
        NOW PRINT.INDIV.LIST GIVEN PT.MAN(BLUE+NN.RED) GO TO 'FINIS'
HERE NOW PRINT.INDIV.LIST GIVEN PT.MAN(RED)
'FINIS' END''OF ROUTINE TO UPDATE TARGET LISTS
```

APPENDIX D. SAMPLE OUTPUT OF THE SIMULATION

In Chapter IV, it is stated that this model conducts a pseudo-simulation of combat. Attached to this appendix, is a sample of the simulation. This output consists of:

1. The initial status chart of all forces and a map of the terrain (in the x-y plane) with the locations of some of the combatants marked.

   a. Appendix A defines the attributes of the combatants listed on the status page.

   b. The notation on the map sheet (Figure D-1) is explained on the bottom of the sheet.

2. The audit trail of soldier number one of force one as he proceeds through the simulation.

   a. Each section of the audit trail defines the location of the combatant in the simulation. Unless marked with the specific routine or event location, the simulation is in the event to detect.

   b. Notation used in the audit trail:

   Z,Z.1,Z.2 - defined in routine PERCENT.VISIBLE

   O.B.D. - observer-building distance

   O.B.A. - observer-building angle (xy-z plane)

   All others self-explanatory.

3. The final status chart of all forces at the end of the simulation.

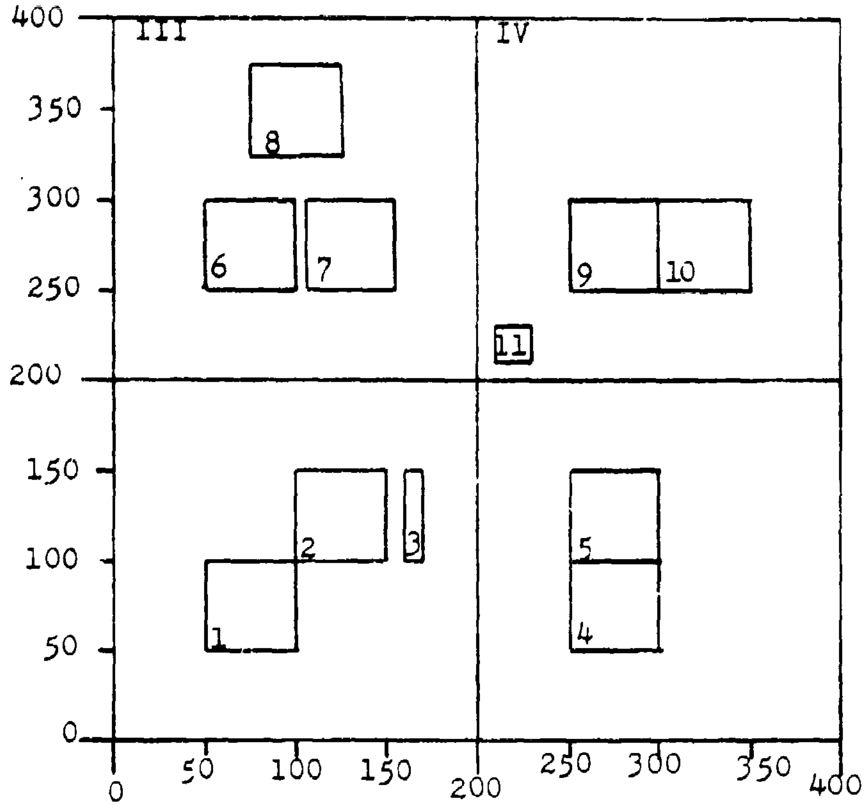
   a. The first chart is the final target list of the Red and Blue forces.

b. The second is the final attribute list of all the forces.

As an aid to the cross-referencing between the audit trail, initial status chart, routines and events, the source codes for the events are contained in Appendix E.

STATUS OF FORCES AT THE START OF THE SIMULATION

MAN	ARMY	POSN	BLK	HSE	FLR	OPEN	DIR	X	Y	Z	ACT	POS
1	1	3	2	4	4	4	4	249.99	76.00	17.00	1	1
2	1	4	2	4	5	999	0	255.00	76.00	21.00	3	3
3	1	4	2	4	5	999	0	260.00	76.00	21.50	2	1
1	2	3	3	8	5	999	0	100.00	326.00	22.00	1	1
2	2	4	3	0	0	999	0	101.00	250.00	.50	4	4
3	2	4	3	0	0	999	0	102.00	251.00	1.00	3	2
4	2	4	3	6	3	0	0	76.00	251.00	12.00	1	1
5	2	3	3	8	4	0	0	124.00	348.00	17.00	1	1
6	2	4	3	8	4	0	0	124.00	350.00	17.00	1	1
7	2	4	3	8	4	0	0	124.00	352.00	17.00	1	1
8	2	3	3	8	4	0	0	101.00	326.00	17.00	1	1



NOTES: (1) Block designations are in Roman numerals.

(2) Building designations are in arabic numerals.

(3) The positions of the Blue and Red forces are not indicated.

Figure D-1. Map of the Urban Terrain

AUDIT TRAIL OF RED SOLDIER #1

```
FROM ROUTINE PERCENT.VISIBLE
          MAN  1 / 1     VS   MAN  5 / 2     BUILDING  7
Z =    -.0100          Z.1 =      .0100      Z.2 =    2.6527
O.B.D = 226.0055       O.B.A =   -.0088
            MIN.F( 1.3313 , 1.0000) =  1.0000 = PCT.VIS

FRCM ROUTINE PERCENT.VISIBLE
          MAN  1 / 1     VS   MAN  7 / 2     BUILDING  7
Z =    -.0100          Z.1 =      .0100      Z.2 =    2.6527
O.B.D = 228.7453       O.B.A =   -.0087
            MIN.F( 1.3313 , 1.0000) =  1.0000 = PCT.VIS

FRCM ROUTINE PERCENT.VISIBLE
          MAN  1 / 1     VS   MAN  7 / 2     BUILDING  8
Z =    -.0100          Z.1 =      .0100      Z.2 =    1.0080
O.B.D = 300.9883       O.B.A =   -.0033
            MIN.F(  .5090 , 1.0000) =   .5090 = PCT.VIS

MAN   1 / 1     HAS LINE OF SIGHT TO      MAN  7 / 2
          AT TIME =    5.1421
          RANGE.TO.TARGET = 303.3965    PER.CENT.VISIBLE =  .5090
          DETECTION WILL OCCUR AT    17.9734

MAN   1 / 1     HAS LINE OF SIGHT TO      MAN  2 / 2
          AT TIME =    6.4628
          RANGE.TO.TARGET = 229.6655    PER.CENT.VISIBLE = 1.0000
          DETECTION WILL OCCUR AT    13.3950

FROM ROUTINE PERCENT.VISIBLE
          MAN  1 / 1     VS   MAN  6 / 2     BUILDING  7
Z =    -.0100          Z.1 =      .0100      Z.2 =    2.6527
O.B.D = 227.3743       O.B.A =   -.0088
            MIN.F( 1.3313 , 1.0000) =  1.0000 = PCT.VIS

FRCM ROUTINE PERCENT.VISIBLE
          MAN  1 / 1     VS   MAN  1 / 2     BUILDING  7
Z =   -5.0000          Z.1 =     5.0000      Z.2 =    2.8738
O.B.D =-202.9135       O.B.A =  -3.1317
            MIN.F( 3.9369 , 1.0000) =  1.0000 = PCT.VIS

FRCM ROUTINE PERCENT.VISIBLE
          MAN  1 / 1     VS   MAN  1 / 2     BUILDING  8
Z =   -5.0000          Z.1 =     5.0000      Z.2 =    3.0123
O.B.D =-290.3762       O.B.A =   3.1313
            MIN.F(  .9939 , 1.0000) =   .9939 = PCT.VIS
```

```
MAN  1 /  1    HAS LINE OF SIGHT TO     MAN  1 /  2
     AT TIME =   6.7945
     RANGE.TO.TARGET = 291.5852    PER.CENT.VISIBLE =  .9939
     DETECTION WILL OCCUR AT   11.3032

FROM ROUTINE PERCENT.VISIBLE
     MAN  1 /  1     VS   MAN  4 /  2    BUILDING  6
Z =   5.0000         Z.1 =    5.0000     Z.2 =   6.0346
O.B.D =-245.3641     O.B.A =  -3.1171
          MIN.F(  .5173 , 1.0000) =   .5173 = PCT.VIS

MAN  1 /  1    HAS LINE OF SIGHT TO     MAN  4 /  2
     AT TIME =   7.7432
     RANGE.TO.TARGET = 246.8249    PER.CENT.VISIBLE =  .5173
     DETECTION WILL OCCUR AT    9.8424

FROM ROUTINE PERCENT.VISIBLE
     MAN  1 /  1     VS   MAN  8 /  2    BUILDING  7
Z =   -.0100         Z.1 =     .0100     Z.2 =   2.8737
O.B.D =-202.5563     O.B.A =  -3.1317
          MIN.F( 1.4419 , 1.0000) =  1.0000 = PCT.VIS

FROM ROUTINE PERCENT.VISIBLE
     MAN  1 /  1     VS   MAN  8 /  2    BUILDING  8
Z =   -.0100         Z.1 =     .0100     Z.2 =   1.0041
O.B.D =-289.8650     O.B.A =  -3.1381
          MIN.F(  .5070 , 1.0000) =   .5070 = PCT.VIS

MAN  1 /  1    HAS LINE OF SIGHT TO     MAN  8 /  2
     AT TIME =   8.1101
     RANGE.TO.TARGET = 291.0291    PER.CENT.VISIBLE =  .5070
     DETECTION WILL OCCUR AT   21.0624

FROM ROUTINE PERCENT.VISIBLE
     MAN  1 /  1     VS   MAN  4 /  2    BUILDING  6
Z =   5.0000         Z.1 =    5.0000     Z.2 =   6.0346
O.B.D =-245.3641     O.B.A =  -3.1171
          MIN.F(  .5173 , 1.0000) =   .5173 = PCT.VIS

ENTERING ROUTINE TO UPDATE.TGT.LIST AT TIME =   9.8424
RED =  1        BLUE =  4       YES.NO =  1    WHO.OBS =  1
     LIST EMPTY AND DETECTION = YES
```

```
              RED ARMY
MAN         DIMENSION OF TARGET ARRAY
 1                       1
                 4

AT   9.8424        1 / 1   DETECTED   4 /  2
     AND WILL SHOOT AT HIM AT  13.8373

MAN   1 / 1    HAS LINE OF SIGHT TO     MAN  3 /  2
      AT TIME =     9.8883
      RANGE.TO.TARGET = 229.7434    PER.CENT.VISIBLE = 1.0000
      DETECTION WILL OCCUR AT    12.2342

FROM ROUTINE PERCENT.VISIBLE
      MAN  1 /  1     VS   MAN  1 /  2    BUILDING  7
Z =  -5.0000          Z.1 =    5.0000     Z.2 =    2.8738
O.B.D =-202.9135      O.B.A =  -3.1317
           MIN.F( 3.9369 , 1.0000) =  1.0000 = PCT.VIS

FROM ROUTINE PERCENT.VISIBLE
      MAN  1 /  1     VS   MAN  1 /  2    BUILDING  8
Z =  -5.0000          Z.1 =    5.0000     Z.2 =    3.0123
O.B.D =-290.3762      O.B.A =   3.1313
           MIN.F(  .9939 , 1.0000) =   .9939 = PCT.VIS

ENTERING ROUTINE TO UPDATE.TGT.LIST AT TIME =  11.3032
RED =  1       BLUE =  1       YES.NO =  1    WHO.OBS =  1
      DETECTED MAN IS NOT ON THE LIST


              RED ARMY
MAN         DIMENSION OF TARGET ARRAY
 1                       2
                 4
                 1

AT 11.3032         1 / 1   DETECTED   1 /  2
     AND WILL SHOOT AT HIM AT  14.9602
```

```
NOTE:  BLUE 4 KILLED BY ANOTHER REC FORCE

ENTERING ROUTINE TO UPDATE.TGT.LIST AT TIME =  12.4627
RED =  1          BLUE =  4        YES.NO =  0     WHO.OBS =  1
     BLUE IS BEING REMOVED FROM REDS LIST


           RED ARMY
MAN       DIMENSION OF TARGET ARRAY
  1                     1
           1

AT  12.4627    SOLDIER    4 /  2  WAS KILLED
TOTAL RED KILLED =   0    VS   TOTAL BLUE KILLED =   2

ENTERING ROUTINE TO UPDATE.TGT.LIST AT TIME =  13.3950
RED =  1          BLUE =  2        YES.NO =  1     WHO.OBS =  1
     DETECTED MAN IS NOT ON THE LIST

           RED ARMY
MAN       DIMENSION OF TARGET ARRAY
  1                     2
           1
           2

AT 13.3950       1 /  1   DETECTED   2 /  2
     AND WILL SHOOT AT HIM AT  16.2091

ENTERING EVENT KILL.MAN AT  14.9602       1 /  1  SHOOTING   1 /  2
     RED =  1       BLUE =  1       RED.LOS.ARRAY =  1

ENTERING ROUTINE TO UPDATE.TGT.LIST AT TIME =  14.9602
RED =  1          BLUE =  1        YES.NO =  0     WHO.OBS =  1
     BLUE IS BEING REMOVED FROM REDS LIST


           RED ARMY
MAN       DIMENSION OF TARGET ARRAY
  1                     1
           2

AT  14.9602    SOLDIER    1 /  2  WAS KILLED
TOTAL RED KILLED =   1    VS   TOTAL BLUE KILLED =   4

ENTERING EVENT KILL.MAN AT  16.2091       1 /  1  SHOOTING   2 /  2
      2 /  2 IS SPARED
      1 /  1 WILL TRY TO REDETECT  2 /  2 AT TIME =  18.2091
      2 /  2 WILL TRY TO DETECT  1 /  1 AT TIME =  20.4587
```

```
FRCM ROUTINE PERCENT.VISIBLE
     MAN  1 /  1      VS   MAN   7 /  2      BUILDING  7
Z =    -.0100         Z.1 =      .0100       Z.2 =    2.6527
O.B.D = 228.7453      O.B.A =    -.0087
          MIN.F( 1.3313 , 1.0000) =   1.0000 = PCT.VIS

FRCM ROUTINE PERCENT.VISIBLE
     MAN  1 /  1      VS   MAN   7 /  2      BUILDING  8
Z =    -.0100         Z.1 =      .0100       Z.2 =    1.0080
O.B.D = 300.9883      O.B.A =    -.0033
          MIN.F(   .5090 , 1.0000) =    .5090 = PCT.VIS

ENTERING ROUTINE TO UPDATE.TGT.LIST AT TIME =  17.9734
RED =  1         BLUE =   7        YES.NO =  1    WHC.OBS =  1
     DETECTED MAN IS NOT ON THE LIST


             RED ARMY
MAN        DIMENSION OF TARGET ARRAY
  1                        2
                 2
                 7

AT 17.9734       1 /  1   DETECTED   7 /  2
     AND WILL SHOOT AT HIM AT  21.3785

AT 18.2091       1 /  1   DETECTED   2 /  2
     AND WILL SHOOT AT HIM AT  21.5616

ENTERING EVENT KILL.MAN AT  21.3785      1 /  1  SHOOTING   7 /  2
     RED =  1        BLUE =   7       RED.LOS.ARRAY =  1

ENTERING ROUTINE TO UPDATE.TGT.LIST AT TIME =  21.3785
RED =  1         BLUE =   7        YES.NO =  0    WHO.OBS =  1
     BLUE IS BEING REMOVED FROM REDS LIST


             RED ARMY
MAN        DIMENSION OF TARGET ARRAY
  1                        1
                 2

AT  21.3785     SOLDIER    7 /  2  WAS KILLED
TOTAL RED KILLED =   2   VS   TOTAL BLUE KILLED =   5

ENTERING EVENT KILL.MAN AT  21.5616      1 /  1  SHOOTING   2 /  2
     RED =  1        BLUE =   2       RED.LOS.ARRAY =  1
```

```
ENTERING ROUTINE TO UPDATE.TGT.LIST AT TIME =  21.5616
RED =  1        BLUE =  2       YES.NO =  0    WHO.OBS =  1
    RED HAS LOST LINE-OF-SIGHT TO BLUE


         RED ARMY
MAN     DIMENSION OF TARGET ARRAY
  1  HAS NOT DETECTED ANYONE
```

STATUS OF FORCES AT THE END OF THE SIMULATION

```
              RED ARMY
MAN         DIMENSION OF  TARGET ARRAY
  1  HAS NO ONE ON HIS TARGET LIST
  2  HAS BEEN KILLED
  3  HAS BEEN KILLED

             BLUE ARMY
MAN         DIMENSION OF  TARGET ARRAY
  1  HAS BEEN KILLED
  2  HAS BEEN KILLED
  3  HAS BEEN KILLED
  4  HAS BEEN KILLED
  5  HAS NO ONE ON HIS TARGET LIST
  6  HAS NO ONE ON HIS TARGET LIST
  7  HAS BEEN KILLED
  8  HAS BEEN KILLED
```

MAN	ARMY	POSN	BLK	HSE	FLR	OPEN	DIR	X	Y	Z	ACT	POS
1	1	3	2	4	4	4	4	249.99	76.00	17.00	1	1
2	1	4	2	4	5	999	0	255.00	76.00	21.00	0	3
3	1	4	2	4	5	999	0	260.00	76.00	21.50	0	1
1	2	3	3	8	5	999	0	100.00	326.00	22.00	0	1
2	2	4	3	0	0	999	0	101.00	250.00	.50	0	4
3	2	4	3	0	0	999	0	102.00	251.00	1.00	0	2
4	2	4	3	6	3	0	0	76.00	251.00	12.00	0	1
5	2	3	3	8	4	0	0	124.00	348.00	17.00	1	1
6	2	4	3	8	4	0	0	124.00	350.00	17.00	1	1
7	2	4	3	8	4	0	0	124.00	352.00	17.00	0	1
8	2	3	3	8	4	0	0	101.00	326.00	17.00	0	1

APPENDIX E: SOURCE CODES FOR THE EVENTS

Attached to this appendix are the source codes for all the events in the urban combat model. Additionally the subroutine DISMTD.CARDIO is attached; although named identically as the routine used in the STAR model, it is not to be confused with the routine in the STAR combat model.

The events and routine are attached in this order:

1. Event STEP.TIME
2. Event LOSCHECK
3. Routine DISMTD.CARDIO
4. Event DETECT
5. Event KILL.MAN

```
*******************************************************************
* EVENT STEP.TIME - SCHEDULES CHECK LOS TIMES FOR EACH SOLDIER *
*******************************************************************

UPON STEP.TIME

DEFINE IRED AND JBLUE AS INTEGER VARIABLES

LET LOS.TIME=0.0
FOR IRED=1 TO NN.RED WITH (ACTIVITY(PT.MAN(IRED)) NE 0), DO
FOR JBLUE=1 TO NN.BLUE WITH ACTIVITY(PT.MAN(JBLUE+NN.RED)) NE 0, DO
    LET LOS.TIME=UNIFORM.F(0.0,15.0,3)
    SCHEDULE A LOSCHECK GIVEN PT.MAN(IRED) AND PT.MAN(JBLUE+NN.RED)
        AT TIME.V+LOS.TIME
    LET LOS.TIME=UNIFORM.F(0.0,15.0,7)
    SCHEDULE A LOSCHECK GIVEN PT.MAN(JBLUE+NN.RED) AND PT.MAN(IRED)
        AT TIME.V+LOS.TIME
    LOOP
LOOP

SCHEDULE A STEP.TIME IN 30. UNITS

END''OF ROUTINE TO SCHEDULE DETECTION AND ATTRITION TIMES
```

```
********************************************************
* ROUTINE TO DETERMINE DETECTION TIMES USING CARDIO SEARCH *
********************************************************

1  ROUTINE FOR DISMTD.CARDIO GIVEN A,B,R,PCT.VIS,X YIELDING DET.TIME

'' A = OBSERVER      B = TARGET        R = RANGE.TO.TARGET
'' PCT.VIS = PER.CENT.VISIBLE          X = UNIFORM(0,1) R.V.
2  DEFINE A,B AND P.HAT AS INTEGER VARIABLES

3  LET MEAN.TIME.DETECT = 2.0-EXP.F(-R*PCT.VIS)
4  LET NDT=NORMAL.F(MEAN.TIME.DETECT,.81,1)

5  LET DET.TIME = EXP.C**NDT

6  END''OF ROUTINE TO DETERMINE DET.TIME
```

```
  ************************************************************
  * EVENT TO CHECK LINES OF SIGHT AND SCHEDULE A DETECTION *
  ************************************************************

 1   UPON LOSCHECK(IOBS,JTGT)

 2   DEFINE IOBS,JTGT,I,J,YES.NO AND WHO.OBS AS INTEGER VARIABLES

 3   IF (ACTIVITY(IOBS) EQ 0 ) OR (ACTIVITY(JTGT) EQ 0) GO TO 'END'
 4   ELSE NOW CHECK.LOS GIVEN IOBS AND JTGT YIELDING YES.NO

 5   IF YES.NO NE 1      GO TO 'END'     ELSE
 6   LET RN=UNIFORM.F(0.,1.,8)
 7   CALL DISMTD.CARDIO GIVEN IOBS,JTGT,RANGE.TO.TARGET,PER.CENT.VISIBLE
          AND RN YIELDING DET.TIME
 8   IF ARMY(IOBS) EQ 1
 9        LET I=TITLE(IOBS)    LET J=TITLE(JTGT)    JUMP AHEAD     ELSE
10        LET I=TITLE(JTGT)    LET J=TITLE(IOBS)
11   HERE
12   SCHEDULE A DETECT GIVEN IOBS,JTGT,I AND J AT TIME.V+DET.TIME

13   'END'
14   END''OF EVENT TO SCHEDULE DETECTION TIMES
```

```
*****************************************************
* EVENT TO DETECT TARGETS AND UPDATE TARGET LISTS *
*****************************************************

UPON DETECT(IOBS,JTGT,I,J)

DEFINE IOBS,JTGT,I,J,YES.NO AND WHO.OBS AS INTEGER VARIABLES

IF (ACTIVITY(IOBS) EQ 0) OR (ACTIVITY(JTGT) EQ 0) GO TO 'END'
ELSE NOW CHECK.LOS GIVEN IOBS AND JTGT YIELDING YES.NO
GO TO ONE,TWO PER ARMY(IOBS)

'TWO'IF YES.NO EQ BLUE.LOS.ARRAY(J,I)   JUMP AHEAD
     ELSE NOW UPDATE.TGT.LISTS GIVEN I,J,YES.NO,ARMY(IOBS)
     LET BLUE.LOS.ARRAY(J,I)=YES.NO     JUMP AHEAD

'ONE'IF YES.NO EQ RED.LOS.ARRAY(J,I)  JUMP AHEAD
     ELSE NOW UPDATE.TGT.LISTS GIVEN I,J,YES.NO,ARMY(IOBS)
     LET RED.LOS.ARRAY(J,I)=YES.NO

HERE       IF YES.NO NE 1      GO TO 'END'      ELSE
LET RN=UNIFORM.F(2.,4.,3)
SCHEDULE A KILL.MAN GIVEN JTGT AND IOBS AT TIME.V + RN

'END' END''OF EVENT TO CHECK LINES OF SIGHT
```

```
*****************************************************************
*  EVENT TO KILL A SOLDIER (50-50 CHANCE) AND UPDATE BLDG.LIST AND *
*       RED(BLUE).LOS.ARRAY AND THE ASSOCIATED TARGET LISTS        *
*****************************************************************

EVENT KILL.MAN GIVEN PTMAN AND GUNMAN

DEFINE GUNMAN,PTMAN AND I AS INTEGER VARIABLES

IF (ACTIVITY(PTMAN) EQ 0) OR (ACTIVITY(GUNMAN) EQ 0) GO TO 'END'

ELSE LET RN=UNIFORM.F(0.,1.,6)
IF RANGE.TO.TARGET GT 300.0   LET RNMAX=.1    JUMP AHEAD    ELSE
IF RANGE.TO.TARGET GT 200.0   LET RNMAX=.5    JUMP AHEAD    ELSE
IF RANGE.TO.TARGET GT 100.0   LET RNMAX=.7    JUMP AHEAD    ELSE
IF RANGE.TO.TARGET GT  50.0   LET RNMAX=.9    JUMP AHEAD    ELSE
                              LET RNMAX=.95

HERE      IF RN LE RNMAX    GO TO 'KILL'    ELSE

     SCHEDULE A DETECT GIVEN GUNMAN,PTMAN,TITLE(GUNMAN),TITLE(PTMAN)
          AT TIME.V + 2.0
     LET RN=UNIFORM.F(3.0,5.0,10)
     SCHEDULE A DETECT GIVEN PTMAN,GUNMAN,TITLE(PTMAN),TITLE(GUNMAN)
          AT TIME.V + RN
     GO TO 'END'

'KILL'    IF MANS.HOUSE(PTMAN) EQ 0        JUMP AHEAD     ELSE
     REMOVE PTMAN FROM BLDG.LIST(PT.HOUSE(MANS.HOUSE(PTMAN)))

HERE      LET ACTIVITY(PTMAN)=0

     IF ARMY(PTMAN) EQ 1 GO TO 'BLUEOBS'     ELSE

ADD 1 TO DEAD.BLUE
FOR I=1 TO NN.RED WITH (RED.LOS.ARRAY(TITLE(PTMAN),I) EQ 1)
          AND (ACTIVITY(PT.MAN(I)) NE 0), DO
     NOW UPDATE.TGT.LISTS GIVEN I,TITLE(PTMAN),0,1
     LET RED.LOS.ARRAY(TITLE(PTMAN),I)=0
     LET BLUE.LOS.ARRAY(TITLE(PTMAN),I)=0
LOOP
     LET BLUE.TGT.LIST(*)=PERS.BLUE.LIST(1,TITLE(PTMAN))
     RELEASE BLUE.TGT.LIST(*)
     LET PERS.BLUE.LIST(1,TITLE(PTMAN))=0
GO TO 'PRINT'

'BLUEOBS' ADD 1 TO DEAD.RED
FOR I=1 TO NN.BLUE WITH (BLUE.LOS.ARRAY(I,TITLE(PTMAN)) EQ 1)
```

```
     AND (ACTIVITY(PT.MAN(I+NN.RED)) NE 0), DO
     NOW UPDATE.TGT.LISTS GIVEN TITLE(PTMAN),I,0,2
     LET BLUE.LOS.ARRAY(I,TITLE(PTMAN))=0
     LET RED.LOS.ARRAY(I,TITLE(PTMAN))=0
LOOP
     LET RED.TGT.LIST(*)=PERS.RED.LIST(1,TITLE(PTMAN))
     RELEASE RED.TGT.LIST(*)
     LET PERS.RED.LIST(1,TITLE(PTMAN))=0

'PRINT'
IF (DEAD.RED GE NN.RED) OR (DEAD.BLUE GE NN.BLUE)
     SCHEDULE A SIM.STOP NOW       ALWAYS

'END'END''OF EVENT TO ATTRIT
```

LIST OF REFERENCES

1. Carpenter, Howard J. and Thurman, Edward E., Parametric Simulation of Infantry Tactics and Equipment (Dismounted STAR), M.S. Thesis, Naval Postgraduate School, Monterey, California, June 1980.

2. Hartman, James K., Parametric Terrain and Line of Sight Modelling in the STAR Combat Model, Naval Postgraduate School Technical Report NPS55-79-018, Monterey, California, August 1979.

3. Ketron Incorporated, Gaming Models for Military Operations in Built-Up Areas, Final Technical Report, KFR 93-76, November 1976.

4. Ketron Incorporated, Gaming Models for Military Operations in Built-Up Areas, Volume I, KFR 102-77, undated.

5. Long, Robert B. and Manata, Jack P., Urban Warfare, Technical Report, Defense Technical Information Center, Defense Logistics Agency, Cameron Station, Alexandria, Virginia, 15 October 1975.

6. USADC Report, Army Small Arms Requirements Study (ASARS) Battle Model, Volumes IIA, IIB, III, 1972.

BIBLIOGRAPHY

Department of the Army, FM 30-102, Opposing Forces Europe, 18 November 1977.

Department of the Army, FM 71-1, The Tank and Mechanized Infantry, 1977.

Hartman, James K., Ground Movement Modelling in the Star Combat Model, Naval Postgraduate School, Technical Report NPS55-80-021, Monterey, California, May 1980.

Needels, Chris J., Parameterization of Terrain in Army Combat Models, M.S. Thesis, Naval Postgraduate School, Monterey, California, March 1976.

Parry, S. H. and Kelleher, E. P., Tactical Parameters and Input Requirements for the Ground Component of the STAR Combat Model, Naval Postgraduate School Technical Report NPS55-79-023, Monterey, California, October 1979.

SPI, Cityfight, Modern Combat in the Urban Environment, Simulations Pbulications Inc., New York, New York, 1979.

United States Army Infantry School, ST 90-10, An Infantry Commander's Guide for Military Operations on Urbanized Terrain (MOUT), Fort Benning, Georgia, November 1978.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4.	Chief TRADOC Research Element Monterey Naval Postgraduate School Monterey, California 93940	1
5.	Associate Professor James K. Hartman, Code 55Hh Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6.	Associate Professor S. H. Parry, Code 55Py Department of Operations Research Naval Postgraduate School Monterey, California 93940	5
7.	Headquarters US Army Training and Doctrine Command ATTN: Director, Studies and Analysis Directorate Mr. S. Goldberg Fort Monroe, Virginia 23651	1
8.	Associate Professor A. L. Schoenstadt, Code 53Zh Department of Mathematics Naval Postgraduate School Monterey, California 93940	1

9. Office of the Commanding General 1
US Army Training and Doctrine Command
ATTN: General Glenn Otis
Fort Monroe, Virginia 23651

10. Headquarters 1
US Army Training and Doctrine Command
ATTN: ATCG-T (BG Morelli)
Fort Monroe, Virginia 23651

11. Office of the Commanding General 1
US Readiness Command
ATTN: General Donn A Starry
MacDill AFB, Florida 33621

12. Headquarters 1
US Army Training and Doctrine Command
ATTN: Director, Maneuver Directorate
Combat Developments
LTC John VanZant
Fort Monroe, Virginia 23651

13. Mr. Walter Hollis 1
Deputy Under Secretary of the Army
(Operations Research)
Department of the Army, The Pentagon
Washington, D. C. 20310

14. LTG Howard Stone 1
Commanding General
US Army Combined Arms Center
Fort Leavenworth, Kansas 66027

15. Director 1
Combined Arms Combat Development Activity
ATTN: ATZL-CAC-A (Mr. Lee Pleger)
Fort Leavenworth, Kansas 66027

16. Director, BSSD 1
Combined Arms Training Development
Activity
ATTN: ATZLCA-DS
Fort Leavenworth, Kansas 66027

17. Director 1
Combat Analysis Office
ATTN: Mr. Kent Pickett
US Army Combined Arms Center
Fort Leavenworth, Kansas 66027

18. Command and General Staff College 1
ATTN: Education Advisor
Room 123, Bell Hall
Fort Leavenworth, Kansas 66027

19. Dr. Wilbur Payne, Director 1
US Army TRADOC Systems Analysis Activity
White Sands Missile Range, New Mexico 88002

20. Headquarters, Department of the Army 1
Office of the Deputy Chief of Staff for
Operations and Plans
ATTN: DAMO-2D
Washington, D. C. 20310

21. Commander 1
US Army Concepts and Analysis Agency
ATTN: MOCA-WG (LTC Earl Darden)
8120 Woodmont Avenue
Bethesda, Maryland 20014

22. Director 1
US Army Material Systems Analysis Activity
ATTN: DRXSY-CM (Mr. Bill Niemeyer)
Aberdeen Proving Grounds, Maryland 21005

23. Director 1
US Army Night Vision and Electro-Optical
Lab
ATTN: DEL-NV-VI (Mr. Frank Shields)
Fort Belvoir, Virginia 22060

24. Director 1
USATRASANA
ATTN: Mr. Ray Heath
White Sands Missile Range, New Mexico 88002

25. Commander 1
MICOM
ATTN: DRSMI-YC (MAJ Hagewood)
Redstone Arsenal, Alabama 35809

26. Director 1
Combat Developments, Studies Division
ATTN: MAJ W. Scott Wallace
US Army Armor Agency
Fort Knox, Kentucky 40121

27. Commandant 1
US Army Field Artillery School
ATTN: ATSF-MBT (CPT Steve Starner)
Fort Sill, Oklahoma 73503

28. Director 1
Combat Developments
US Army Aviation Agency
Fort Rucker, Alabama 36362

29. Director 1
Combat Developments
US Army Infantry School
Fort Benning, Georgia 31905

30. Director 1
Armored Combat Vehicle Technology
Program
ATTN: COL Fleming
US Army Armor Center
Fort Knox, Kentucky 40121

31. Director 1
Combat Developments
ATTN: MAJ William D. Meiers
US Army Air Defense Agency
Fort Bliss, Texas 79905

32. Commander 1
US Army Logistics Center
ATTN: ATCL-OS (Mr. Cammeron/CPT Schuessler)
Fort Lee, Virginia 23801

33. Commander 1
USAMMCS
ATTN: ATSK-CD-CS (Mr. Lee/Mr. Marmon)
Redstone Arsenal, Alabama 35809

34. Director 1
US Army AMSAA
ATTN: DRXSY-AA (Mr. Tom Coyle)
Aberdeen Proving Grounds, Maryland 21005

35. Office of the Deputy Chief of Staff 1
for Combat Developments
US Army TRADOC
ATTN: Major General Dick Boyle
Fort Monroe, Virginia 23651

36. Deputy Commanding General 1
Combined Arms Combat Development Activity
ATTN: ATZL-CA-DC
Fort Leavenworth, Kansas 66027

37. Commandant 1
US Army Signal School
Fort Gordon, Georgia 30905

38. CPT Bradford G. Loo 1
1363 East 52d Street
Brooklyn, New York 11234